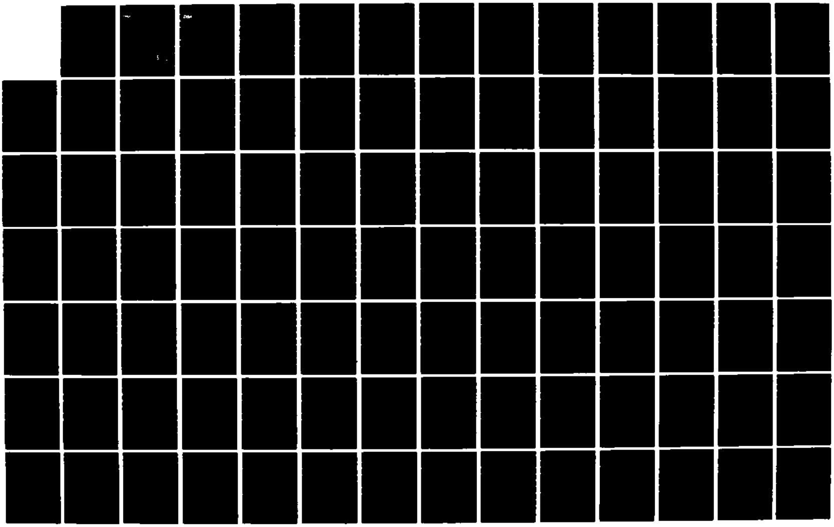
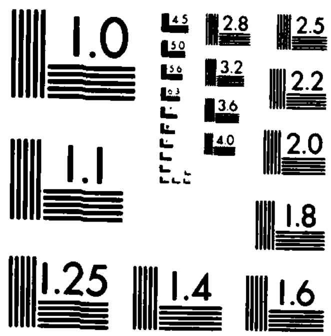
AD-A150 424 ACQUISITION AND LOGISTICS INFORMATION AND ANALYSIS SYSTEM (ALIAS) USER'S GUIDE(U) DECISION-SCIENCE APPLICATIONS INC ARLINGTON VA M S CAREY ET AL. 31 OCT 84 DSA-618 N00014-82-C-0813 1/4

UNCLASSIFIED F/G 15/5 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

13

DSA DECISION·SCIENCE APPLICATIONS

DSA Report #618

October 31, 1984

AD-A150 424

ALIAS USER'S GUIDE

DTIC ELECTE FEB 21 1985 S D E

DTIC FILE COPY

Submitted to:

Scientific Officer
Naval Center for Acquisition Research
NAVMAT 08
Washington, D.C. 20360

Attention: Dr. Thomas C. Varley

This document has been approved for public release and sale; its distribution is unlimited.

85 02 20 004

Suite 1000, 1901 North Moore Street, Arlington, Virginia 22209 (703) 243-2500

DSA DECISION-SCIENCE APPLICATIONS

DSA Report #618

October 31, 1984

ALIAS USER'S GUIDE

M.S. CAREY
J.C. KRUPP
J.M. KABAT

Distribution unlimited per Dr. Thomas C. Varley, Navy Office for Acquisition Research, ATTN: NAVMAT-08, Washington, DC 20360

Submitted to:

Scientific Officer
Naval Center for Acquisition Research
NAVMAT 08
Washington, D.C. 20360

Attention: Dr. Thomas C. Varley

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS N/A
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Direct (Naval Sea Systems Command(SEAOOD)) Washington, D.C. 20362
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) DSA Report No. 618	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Decision-Science Applications, Inc. 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Navy Office for Acquisition Research
6c. ADDRESS (City, State and ZIP Code) 1901 North Moore Street, Suite 1000 Arlington, Virginia 22209	7b. ADDRESS (City, State and ZIP Code) NAVMAT 08 Washington D.C. 20362
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable) ONR	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-82-C-0813
8c. ADDRESS (City, State and ZIP Code) Department of the Navy 800 North Quincy Street Arlington, Virginia 22217	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: N68342 PROJECT NO.: 980770 TASK NO.: WORK UNIT NO.:
11. TITLE (Include Security Classification) ALIAS USERS GUIDE Volumes I and II	

12. PERSONAL AUTHOR(S)
M. Carey, J. Krupp, J. Kabat

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM 10/82 TO 8/84	31 October 1984	

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	ALIAS, DATA BASE, DBMS, MODELS, STRUCTURED PROGRAMMING

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This documentation explains the structure of the Acquisition and Logistics Information and Analysis System (ALIAS). With this documentation, the experienced programmer should be able to easily maintain and expand the ALIAS system. In addition, the manuals explain all standards to which ALIAS extensions should conform. For the non-programmer these manuals describe the philosophy of ALIAS and its extent and limitations.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Joseph C. Krupp	22b. TELEPHONE NUMBER (Include Area Code) 703-243-2500 22c. OFFICE SYMBOL DSA

TABLE OF CONTENTS

Section	TABLE OF CONTENTS (Continued)	Page

LIST OF FIGURES

Figure Number	LIST OF FIGURES (Cont'd).	Page

LIST OF TABLES

1.0 INTRODUCTION

Decision-Science Applications, Inc. (DSA) has created, installed, and tested a system for the Navy termed the Acquisition and Logistics Information and Analysis System (ALIAS). This analyst information system is used by NAVSEA 90 to improve ship acquisition planning so that programs are on time, are within budget, and fit cohesively into a fleetwide programming plan. ALIAS provides the Navy with the tools needed to help identify poorly planned programs quickly, and to allow for quick analysis of alternative program schedules. It consists of 1) a large, integrated relational data base, 2) a menu command system which allows the user to perform any task through the use of menu selection or direct command, 3) an extensive help subsystem which guides the novice through the system or provides the experienced user with a quick reference guide, 4) several libraries of utility procedures, and 5) a variety of high level analytical and functional modules which interface with both the command system and the data base. A scenario system is employed to allow any variety of "What if?" questions without the need for multiple copies of the data base ensuring quick turnaround with data integrity and security.

1.1 DESCRIPTION OF THIS MANUAL

This manual is designed to familiarize the new user with the current capabilities of the ALIAS system as implemented on the current host HP-3000 computer, to describe how to exercise these capabilities, and to provide a reference guide for the experienced user. For a complete description of the system architecture and an understanding of how the ALIAS system actually works, the user is referred to the companion volume, The ALIAS Maintenance and Expansion Guide.

1.1.1 Organization of This Manual

The manual describes four things:

1) Basic concepts you will need to know.

2) How the system works in practice. This is communicated by giving an example of an ALIAS session.

3) The logical details of how various parts of the system work.

4) How to use the various parts of the system---what commands need to be given, what data is required, etc.

Section 2 presents all the concepts that you will need to know. Section 3 presents a sample session in a format which features what will appear on screen or printer on the right-hand pages, with running comments on the left-hand pages. You can learn a lot about how to use the system just by looking through Section 3.

Section 4 describes how to operate the ALIAS Command System. Sections 5-7 describe the role of the ALIAS data base and how to access and modify it. Sections 8 and 9 describe the applications modules which ALIAS currently offers.

ALIAS outputs are described as the discussion goes along. One of the characteristics of the system is that the format and content of any given output can vary widely (under your control), so it is impossible to present an exact and complete set of system outputs. If you want to see some sample outputs right away, turn to Section 3 or Sections 8 and 9 and look for Figures.

1.1.2 Related Publications

To find out more about the HP 3000 computer, see Using the HP 3000, Using Files, MPE Commands, and the EDITOR Reference Manual, all by Hewlett-Packard. To find out more about the RELATE DBMS, see Introduction to RELATE and the RELATE Reference Manual (the CREATE and GRAF manuals may also be useful), all by Computer Resources, Inc. (CRI).

The *ALIAS Maintenance Guide* describes the actual ALIAS software structure (considerably more complex than it may appear) and the details of how it works. A knowledge of programming is required to fully understand the *Maintenance Guide*, but non-programming users curious about the software can learn a lot by reading the first several sections of that manual.

1.2 DOCUMENTATION STANDARD

The set of ALIAS *Guides* pay attention to but do not strictly follow the DoD Automated Data Systems Documentation Standards (7935.1-S, September, 1977). The set of manuals exceed the standard in terms of information content, and each is organized somewhat differently in order to make them more useful to ALIAS users and maintenance personnel. If you are accustomed to reading documentation according to the standard, you may see Appendix A of the *Maintenance Guide*, which cross-references sections mandated by the standard to sections of this ALIAS documentation.

1.3 OVERVIEW OF ALIAS STRUCTURE

The design goals for ALIAS software are ease of use, high user control of system operations, output consistency, flexibility, and expandability.

As an analyst or manager, you cannot be more productive if your software tools are difficult to learn and use. On the other hand, you cannot be more effective if they are so simplified that they cannot handle the range of situations you face. There is typically a tradeoff in software between control and usability---ALIAS software circumvents the tradeoff whenever possible, choosing to give you control if forced.

ALIAS minimizes inconsistency, which usually results from the use of different underlying data, by using a single integrated data base for everything. However, ALIAS is not limited to only one study (i.e. one data set) at a time. There can be

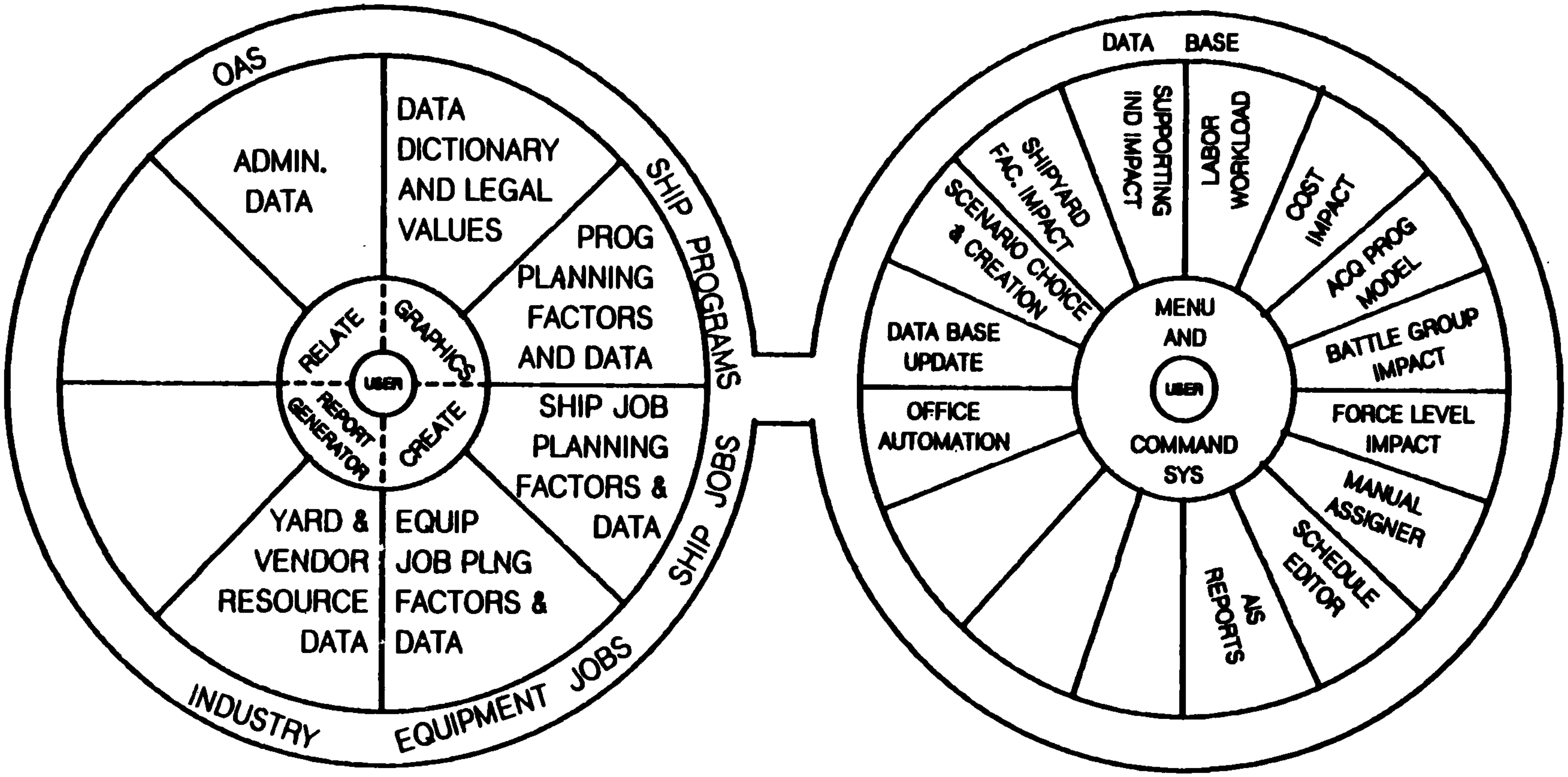
multiple data base partitions (i.e. studies or scenarios) current at once. But any single scenario is always internally consistent

Any tool must be flexible enough to accommodate the range of situations you faced. ALIAS is a modular system, allowing selection and combination of the tools appropriate to any task (with no requirement to step through irrelevant sequences). Further, the individual modules typically have a variety of mode options ("switches") which can be set to customize their operations for a given task.

You will doubtless encounter problems you know ALIAS could help you with ("I need the delivery dates of all ships planned for construction in states with a population of X million or more, by state"), but for which no specific software exists. The dynamic query capabilities offered by ALIAS' DBMS often make it possible for you to construct appropriate software quickly by yourself. For more complex tasks, the internal design of ALIAS and the support offerred by the ALIAS environment will help your programmers create the solution quickly.

Once a solution is created, it can be <u>permanently</u> added to the stock of ALIAS tools (and made publicly available, if desired) with relatively little effort (a familiarity with the material in the <u>Maintenance Guide</u> is required to do it reliably, but given the familiarity the job can usually be done in an hour or less).

ALIAS software is structured in a large number of pieces which work in concert. The pieces can be categorized in a number of ways; in this manual two approaches will be used. Figure 1-1 presents the first, in which ALIAS is presented as being composed of two basic parts: a data base with DBMS and a block of analysis software. The analysis software requires the data base as a source of inputs and (sometimes) as a repository for



DATA BASE MANAGEMENT SYSTEM AND DATA BASE GROUPS

COMMAND SYSTEM AND ANALYSIS MODULES

Figure 1-1. Acquisition and Logistics Information and Analysis System Software Architecture

outputs. The data base requires one of the modules, the Data Base Updating System, if its contents are to be current and consistent. Both the data base and the analysis software can be further broken down: the data base into groups by subject, and within groups into individual tables of data; and the analysis software into individual modules which perform specific tasks. The data and the programs, then, can be thought of as fairly separate entities which are still crucially dependent on one another.

A similar presentation of structure is given in Figure 1-2. This one is more relevant to day-to-day ALIAS use. You will make use of ALIAS in two basic ways: by running the Core command system, and by running the DBMS. The Core command system is where you choose among analysis modules. It is also the gateway to use of the Data Base Updating (DBU) system, which you are likely to use a lot.

Even if you never do data entry as such, the DBU provides a way to find and inspect individual data base items, and to make modifications. ALIAS analysis software is typically very "data-driven" (part of the high-control philosophy); one of the ways you can make it handle special cases is by changing data base values. For example, you can examine the impact on deployable battle groups of changing the official carrier retirement schedule to some alternative by changing the official retirement dates in the data base. You can do this without destroying the official data or disturbing anyone else's work. Changing a dozen dates can be slightly tedious, but not nearly so tedious as having to compute the force structure by hand because software doesn't let you change the underlying data.

The scenario manager (or more precisely, the scenario orientation built into the entire ALIAS structure) is what lets you change those retirement dates without affecting anyone else.

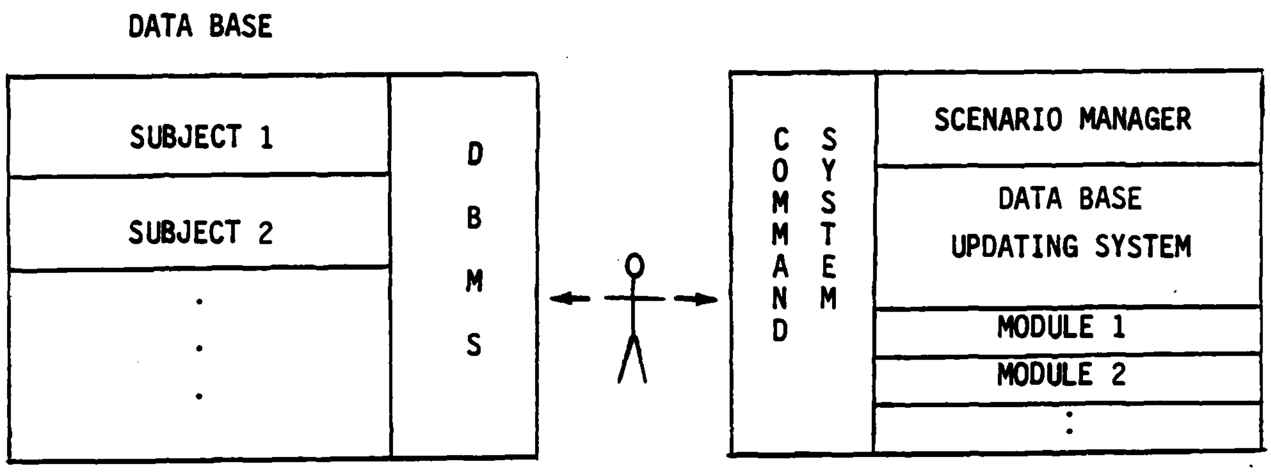


Figure 1-2. ALIAS Structure

Think of a scenario as being your file drawer in a filing cabinet, where the filing cabinet is the data base. You can be changing the contents of your drawer while your neighbor works with hers, and her work will not be disturbed.

The Core command system's main purpose is to present menus listing the choices you have. Although it does allow you to set the values of "parameters" (switches) which control the operation of analysis modules, the command system does not itself generate any productive output. Learning the command system is a prerequisite for working with ALIAS: running the command system isn't what you really want to be doing, but you can't do what you want to do without it. By organizing things this way your work as a whole goes faster, because the command system invisibly handles a lot of overhead work for you.

The second basic means of using ALIAS, by running the DBMS, is for making ad hoc queries and for creating new reports. The DBMS gives you direct access to the data base files, allowing you to combine and consolidate information using relational methods, which are very flexible. However, you will not be allowed to change data base data through the DBMS, only through the DBU.

2.0 CONCEPTS AND TERMINOLOGY

This Section will present the major software concepts you need to understand in order to use ALIAS, and will also define some of the terms used in the rest of the manual. The Section is organized as an alternative introduction to the structure of ALIAS; the concepts and terms are set off to make it easier to refer back to them later. Concepts and terms are in bold face; major definitions are in single-spaced indented paragraphs.

The Section is likely to be difficult to read straight through because it presents many abstract ideas and rules with almost no examples. We suggest that you glance through the Section to get a general idea of what it offers, reading the summary at the end closely. Then go on to Section 3 (the sample session) for examples. Turn back to this Section and read the relevant paragraphs (easily found using the alphabetical list in section 2.1) as necessary as you read the rest of the manual.

Note that you are assumed to be familiar with the terms and concepts of ship acquisition program analysis.

2.1 ALPHABETICAL LIST OF TERMS AND CONCEPTS WITH PAGE NUMBERS

SHOWJOB command 3
structure --- see scenario structure or data base structure
system 4
system core 5
terminal 2
text editor 12

2.2 THE COMPUTER

ALIAS is composed of software running on a general-purpose minicomputer, the "host":

HOST COMPUTER: The computer that ALIAS runs on. Currently the host is a Hewlett-Packard 3000 (HP 3000 or **HP** for short) located at the offices of PMS 392. You will need to know the basics of using an HP 3000 in order to use ALIAS.

You will access this computer using a **terminal**, a device with a keyboard and a TV-like screen. The screen will be referred to in this manual as the **display**. As you type things on the keyboard, a blinking white box or underbar on the display will move along, leaving symbols in its wake. The moving marker is called the **cursor**; it indicates "where you are" on the display.

The terminal's **keyboard** will have several keys in addition to the standard letters and numbers to be found on a typewriter. Among the most important is the **RETURN** or carriage return key. Hitting this key gets the computer's attention, telling it that you are ready for it to process what you have been typing. Also important is the **BREAK** key, usually located at the upper right hand corner of the keyboard. Hitting this key tells the computer to stop whatever it is doing and return you to the operating system.

OPERATING SYSTEM: The operating system is the host computer's supervisory program. It is in control of all of the HP's resources; its function is to allocate them to users and to ensure that no user disturbs the work of another. On the HP the operating system is known as "**MPE**", which stands for "Multi-Programming Executive." You can tell you are at the operating system "level", i.e. you are talking to it directly, when the computer types a ":" and waits for you to type in a command. There

are many operating system commands you can give. Among the most useful are **LISTF** to list the files you have, **PURGE** to delete a file, and **SHOWJOB** to find out who else is using the computer. See HP's manuals to find out more about MPE.

When you see the ":" prompt after hitting the BREAK key you may give the "**ABORT**" command, which will permanently terminate the program you were running. If you want to go on with what you were doing, you may type "**RESUME**".

Another important key is the control key, likely to be labeled "**CTRL**" and located at the left middle of the keyboard. The control key can be thought of as a special sort of SHIFT key. It makes other keys mean things they don't usually mean, just like the shift key can make the "1" mean "!".

There are three particularly useful things you can do using the control key. If something is being typed out on your display that you really don't want to see, but you don't want to go so far as to abort the program that is typing it, you may hold down the control key and the "Y" key simultaneously (referred to as "**CTRL-Y**" or "**CONTROL-Y**"). In most (not all) cases this will cause the rest of the output to be thrown away.

If you want to see all of what is being typed but it is going too fast, you can use "**CTRL-S**" to temporarily stop the output. When ready you can type "**CTRL-Q**" and output will resume where it left off.

Your keyboard is also likely to have some **cursor control keys** or "arrow" keys and a **DEL** or "delete" key. These will not work properly with ALIAS programs. They will move the cursor around the display in most cases, but the programs will not "see" what you see when you finally hit the RETURN key. Do not use the arrow keys. The **BACKSPACE** key (upper middle right area of keyboard) will work, though.

In order to use ALIAS or any facilities of the host computer, you must be **logged on.** Logging on is a process whereby you identify yourself to the computer, are recognized as a valid user, and are assigned resources to work with. See the beginning of Section 3 for guidance on how to log on.

Throughout this manual the term **system** will be used often. In some sense it will always refer to the same thing, that being the combination of the host computer and ALIAS software, but the references will be at many levels of aggregation. For example, ALIAS software is composed of many semi-independent pieces of software, some of which can be referred to as systems in their own right. Whether the word "system" is referring to part or all of ALIAS should always be clear from its context.

2.3 SYSTEM BUILDING BLOCKS

Think of the software portion of ALIAS (which is what this manual is exclusively concerned with) as residing in or on top of the HP hardware and the MPE operating system. The software uses six major building blocks: a **data base**, a **data base management system** (DBMS), a **System Core**, **modules**, **peripheral files**, and **peripheral processors**.

DATA BASE : The ALIAS data base holds a description of part of the real world, in particular of ship acquisition programs and the resources needed to carry them out. When you use ALIAS you are almost always drawing on this world description and/or are changing it to be more up-to-date or to suit your purposes. The data base can be divided into two parts conceptually: one, the **structure**, is the repository or vessel in which the data is kept; the second is the data itself. The term "data base" will usually refer to both. It will become clear to you that the data itself is "living", constantly changing, but be aware that the structure is also quite changeable. Should you need a repository for data not currently available, the ALIAS data base can be expanded to handle it.

DATA BASE MANAGEMENT SYSTEM (DBMS): A DBMS is a supervisor for a data base. It sets up and maintains the structure, and provides means of changing and retrieving the data. By "buffering" all actions with regard to a data base through a DBMS, everyone is guaranteed that there is a single, consistent set of rules for using the data base. ALIAS uses the RELATE DBMS, which provides a query language, report generator, graphics capability, and data-entry form management language. The query language and report generator are most important for an ALIAS user to know about---they provide a set of tools for asking questions about the state of the world (i.e. the contents of the data base) and for receiving the answers in a nice format. It is important to know that RELATE is a stand-alone program on the HP; this means it can be executed independently of the rest of ALIAS. In the "dumbbell" representation of the ALIAS structure in Figure 1-1, note that the data base and DBMS compose the entire left-hand circle.

SYSTEM CORE : The Core is to ALIAS what MPE is to the HP 3000; a supervisory program which provides services to you the user and which ensures that users do not conflict with one another. It does little or no "work" in the sense of producing outputs you are interested in. Its function is to provide an organized means of letting you choose among ALIAS capabilities. If you have made some use of ALIAS already, you can think of the Core as being composed of the Command System, the scenario system, and the DBU.

MODULES : ALIAS modules are processors which concentrate on performing a particular function, usually an analytical function. Most draw upon and/or change the contents of the data base, and most produce some sort of printed output. Modules are public resources likely to be of interest to a wide variety of users. An example of a module is the Force Level Report Generator, which provides estimates of the number of ships which will be deployable in future years.

PERIPHERAL FILES : You may wish to develop your own data bases as parts of analyses that you or a group of your colleagues are involved in. Though not actively supported or supervised by ALIAS system managers, such data bases are considered a part of the resources of ALIAS as a whole. Similarly, ALIAS sometimes expects to find certain kinds of

information in text files which you make up independently.

PERIPHERAL PROCESSORS : Similar to a peripheral file or data base, a peripheral processor is one developed by an ALIAS user for his or her personal use or for the use of a small group. Most such "processors" are DBMS report generation command files (files which cause a custom report to be generated).

In these last definitions some terms were used which have not yet been defined, namely "**files**" and "**processors.**"

The operation of any piece of computer software requires three things: **data** to be manipulated, a **processor** to perform the manipulations, and **instructions** from you the user on what is to be done. Operation always results in **outputs.**

In ALIAS, the data always comes from the data base. You can always look at the data. The data is a picture of the ship acquisition and construction world.

Processors, however, are generally somewhat hidden from you. You typically cause an ALIAS processor to go to work by choosing from among a list of options on a menu. You will often neither know nor need to know much about a processor's nature.

ALIAS expects you to provide **instructions** in several different formats. Most of this manual is concerned with what instructions you must/may provide and with how to provide them.

Outputs may either be changes to the data base, information typed onto your terminal's display, or printed reports.

An analogy may help here. ALIAS is a productive system which involves several parts, somewhat like your kitchen. Say you are going to bake a cake. You will first need to take raw materials from your cabinets (the flour, butter, etc. are analogous to the data in the data base, the cabinets to the data base struc-

ture, and you are the DBMS in this case). Then you will need to perform at least two operations on the materials (data), mixing and baking. Your mixer and your oven are like processors---they act on the materials (data) but are themselves unchanged when the actions are completed. Your instructions are the speed at which you set the mixer, your pushing of its on and off buttons, and your setting of the oven's temperature and timer. Notice that the precise way in which you give the instructions depends on the nature of the appliance (processor). Notice also that the output, the cake, might be placed back in a cupboard (analogous to outputs going back into the data base rather than to a printer), perhaps to serve later as an input to a frosting operation.

FILES : Files are the basic unit of permanent data storage on the HP. They hold both data and processors. There are many different types of files, just as there are many different purposes for them. However, as an ALIAS user, you are likely to be concerned with only two types: text or "editor" files, and RELATE files (data base storage structures).

Files that belong to you will typically be kept in your personal file **group.** The HP computer lumps files into groups for organizational purposes. There are a large number of ALIAS groups, but you will only need to know about those that hold data base relations (files) and about your own personal group. Your personal group will be named after your user name "base". For example, if you have user names JOHNA and JOHNR, then your personal group will be called JOHN.

A **file name** is composed of its name and its group separated by a ".". For example, "MYDATA.MYGROUP" is a file called "mydata" located in the "mygroup" group. When you want to refer to a file (say in a PURGE command) you must give both the file name and group name unless the file is in your personal group, in which case you need give only the file name portion.

2.4 YOUR ENVIRONMENT

As you use ALIAS you may find it helpful to think of your actions as movements within a physical space, e.g. movements in a

house with many rooms. This can be helpful because what you can do will depend on "where you are", i.e. on your **context**. You can only wash clothes in a laundry room; you can only change data in the ALIAS data base using the DBU. If you ever become completely confused about what is happening, the most likely cause is thinking you are in one place when you are actually in another.

Such confusion can occur since you must often look closely to identify your context. The indicator can be as subtle as the character(s) used as the **prompt** (i.e. what the computer prints to tell you it is ready for your next command). The MPE context is identified by the ":" prompt, for example, while the text editors use a "/".

Most ALIAS software is **menu-oriented**, i.e. it presents you with a full display screen of information/choices at a time, making the context somewhat easier to identify. The nature of the information on the display and especially the title at its top are good indicators. For example, you can always tell you are "in" the ALIAS Command System because a line reading "* ALIAS COMMAND SYSTEM *" will appear at the top of its menus.

The sample session in Section 3 will illustrate all of ALIAS' contexts.

Closely related to the notion of context is that of your **environment**. Where context is the room you are in, your environment is the scent of the air you breathe, the kind of music playing in the background, etc. You can change these things, but once set they will remain constant no matter what "room" you are in. Examples of things which are part of your ALIAS environment are the printer your reports come out on and the type of terminal ALIAS thinks you are using.

Your main activity as an ALIAS user will be giving **commands** in response to the various prompts. These commands are the primary **instructions** (as defined in Section 2.2) that ALIAS expects. Commands can have a variety of effects, from displaying a different menu to deleting some data from the data base. The main thing which makes different contexts different is that the commands which you can give (or their effects) are not the same.

Commands can be divided into three basic groups: **choices**, **instructions**, and **settings**. A choice command is one which indicates what it is you want to do next, or which is an answer to an explicit question. Choice commands often change your context.

A **setting** command is one which you use to change a system value, e.g. which printer your output will appear on. The way you change a setting is by telling ALIAS which value you want to change and what you want to change it to. For example, you might set value 3, "your" printer, by "3=DAISY".

An **instruction** is a command which tells the part of the system you are working with to take some action: delete some data, find some data, etc. Instructions and choices are similar; they differ in that instructions implement a change somewhere, or specify how something is to be done.

There are **defaults** for many things in ALIAS. Defaults are variable values or instructions which are used <u>unless you specify otherwise</u>.

Most ALIAS environments are based on **menus**, **screens**, or **pages**. The three are similar in that each takes up the whole display on your terminal. When ALIAS shows a menu, screen, or page to you the first thing it does is blank the display; then it types out the menu/screen/page.

The three differ in the kind of information shown and in the method they expect you to use to give commands and set values. Menus and screens can both present lists of choices for your next action, or can show values you can change. But in menus you are expected to make your choices and settings in the usual line-oriented fashion, while screens are "fill-in-the-blank" oriented.

This terminology may be a little confusing, since "menu" usually refers to a list of choices for your next move, independent of the method you use to specify which choice you want. Although the distinction made here is important, it will always be obvious to you from the contents of the display what you face: if you see the word "COMMAND" near the top of the display next to a bright green or white "blank space" in inverse video, then you are in a screen. If the last thing typed as the display is filled is the prompt "COMMAND:" then you are in a menu. Either could be presenting you with a list of choices to pick from or a set of values for your inspection and modification.

A **page** requires line-oriented responses from you (it will prompt you for commands instead of presenting a fill-in-the-blank area in inverse video). This makes it similar to a menu, but pages present more complicated data such as lists or tables. Often they are so large that they cannot fit on your terminal's display, so you are given the capability to "turn" the page, i.e. to look up or down and/or right or left. **Paging** is the act of changing which segment of the list/table you are looking at.

Let's continue with the notion that use of ALIAS is like moving among the rooms of a building. Each menu, screen, or page is like a room, any of which you are free to enter (if security permits). However, there are limits on how you can move from one room to another in some cases. In particular, menus are always organized in a **hierarchy**. Figure 2-1 pictures a hierarchy, which is like an upside-down tree.

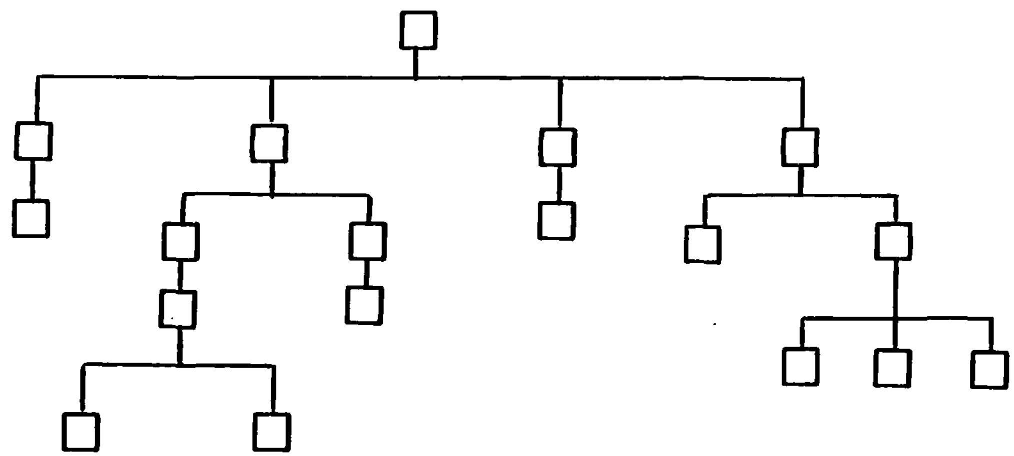


Figure 2-1. A Hierarchy

In moving among menus in a hierarchy, you must always stay on the paths or branches. Thus if you start in the top menu and work down three levels of a branch, you will have to work back up that branch before you can use the menus on another branch (there are shortcut commands that make this easier).

Screens are organized in hierarchies too, but it is often possible to "jump" across branches where they are involved.

Think of the ALIAS Core as a convention center designed to handle several related conferences at once. The only way in is via a central foyer, which opens on several suites devoted to particular conferences. These suites may or may not have hierarchical room organization---you may or may not be able to walk freely from any room to any other without backtracking. However, to step from one conference to another, you must go back through the foyer, the top of a grand hierarchy.

2.5 DATA BASE CONCEPTS

The ALIAS data base is organized along relational lines (rather than network or hierarchical lines). It is composed of **relations**, which are just independent tables with rows and columns of data. These relations/tables are stored in the form of HP files, accessible by use of the RELATE DBMS.

Figure 2-2 shows a sample relation. This one has three rows of five columns each with some construction schedule information. Notice that each row is dedicated to a particular ship, and that each column is for a single piece of information about the ships.

Relations typically can have an unlimited number of rows (i.e. can handle as many ships as you want), but only the specific columns they were created with. Thus if the owner of the sample wanted to start tracking launch dates, he would have to

CLASS	HULL	AWARP	START	DELIVERY
FFG-7	99	01/26/1998	06/01/1998	08/13/2000
DDG-51	81	03/15/1996	07/12/1997	02/01/2001
CVN-68	92	10/15/2010	10/15/2011	06/01/2016

Figure 2-2. A Sample Relation

create a new relation with six columns instead of five and then move his existing data from the sample relation into the new one.

Columns in relations are usually called **fields**, each of which has a **field name.** You can use the field names as part of your RELATE commands to manipulate data. Rows are called tuples or **records.**

Relations always have **key fields.** Key fields are the ones whose values you give when you ask that specific data be extracted from a relation and shown to you. In the sample, the **primary key fields** are CLASS and HULL. The sample relation holds information about individual ships, and the contents of these two fields (columns) identify which ships any given record (row) is for. In this manual, the term **key** will always refer to the primary key fields.

There can be other keys as well. In fact, any field or combination of fields can form the key of a query. For example, we might ask RELATE to tell us which ships recorded in the sample relation were awarded before the year 2000 but delivered after 2000 (we would be told FFG-7:99 and DDG-51:81); in this case, the key is AWARD,DELIVERY.

The **key** or primary key fields are those whose values can be used to uniquely identify a record. If you are an experienced RELATE user you know that RELATE's inner workings are such that a relation need not have a key in this sense, i.e. it may contain duplicate data. However, ALIAS data base relations are created and managed by the Data Base Updating System in ways that ensure there is always a primary or unique key.

RELATE offers a service closely related to keys called indexing. **Indexes** are alternative sort-orderings which can be used for printout of the contents of a relation. The default sort-ordering is just the order in which the data was entered,

which is usually not very useful. For example, we might want to print out the data in the sample relation in Figure 2-2 by ship on some occasions (i.e. by class,hull), but by date of award on other occasions.

Indexes can be either permanent or temporary. A permanent index makes an ordering instantly available, but imposes some overhead on data base updating operations. On the other hand, it may take RELATE several minutes to set up a temporary index.

In thinking about how to use the ALIAS data base it is important to remember that the various files/relations are in fact related to one another even though they are technically completely separate. In particular, they can be merged or joined using the values of similar fields in separate relations. For example, say that in addition to the sample relation in Figure 2-2 there was a second one with the fields CLASS and TONS. Using RELATE's SELECT command, it would be possible to print out a list of ships in the sample, their dates, and their displacements. In fact, much more complex and powerful things can be done, making it possible to answer complicated questions by combining the information in several data base files.

See the RELATE manuals for more information about use of RELATE.

2.6 SCENARIOS

One of the most useful features of ALIAS and the ALIAS data base is its capability to handle multiple studies simultaneously. Two people can be using the system at the same time, one working on a standard POM projection and another on a mobilization exercise, and they will not interfere with one another. They will both be drawing data from the same data relations, but it will be different data; it will often seem to each that he has a dedicated data base.

Exercises or studies are called **scenarios** in ALIAS. A scenario is really a named container for a study's data, and is thus very much like a file drawer. The ALIAS data base can be thought of as an expandable file cabinet which gets a new drawer any time someone creates a new scenario.

When you use the Core you may only work with one scenario at a time, and you must be the only person using it. This keeps users from interfering with one another. In terms of the file drawer analogy, if you have the contents of the drawer on your desk and are making some changes, Joe at the next desk should not be making different changes at the same time. If Joe is working out of another drawer, though, then there is no problem.

A subtle problem with this arrangement has to do with data base updating, the process of keeping the data base current. Must the updating be done for each and every scenario?

Say there are three different POM projection scenarios, all of which need to make use of the latest data about current shipyard loads. The person entering the latest reports from the shipyards doesn't make the entries three times. Instead, regular data updating is done for only one scenario, called "MAIN".

Instead of having their own current yard load data, the POM projection scenarios make use of the current load data from MAIN. This is called **indirect access**, as opposed to **direct access** in which each POM scenario would have its own current load data (which the POM scenario's owners would have to keep up-to-date themselves).

Instead of having file drawers with a complete set of folders, one per subject, the POM scenarios have only a partial set of folders; they "borrow" the current load folder from the MAIN drawer whenever they need current yard load data. This way

they can benefit from the ongoing data updating activity which supports MAIN.

However, in exchange for this "borrowing" privilege, the POM scenario's creators had to agree not to make any changes to MAIN's current yard load data. If one needs to change this data for some reason, then he must go to the copier and make up his own current yard load folder by copying MAIN's. This folder goes in his drawer and he loses his borrowing privilege (at least until he throws the new folder away).

The creator of an ALIAS scenario must decide at creation time which subjects he will borrow (use **indirectly**) from other existing scenarios, and which he will need to make changes to (use **directly** and maintain current himself). As indicated in the analogy, he can change his mind later about these structural decisions.

2.7 MISCELLANEOUS CONCEPTS

FORMAT CONTROL FILE: A format control file is a device which you use to specify the contents and format of a report you want ALIAS to produce. Format control files are currently required by the Force Level and Battle Group report generators. You make them up according to the rules given in Section 9 using one of the text editors.

TEXT EDITOR: This is a program which permits you to create and change files of text. The editors on the HP are line-oriented editors, meaning that they only let you work with one line of text at a time. The best editor on the HP is called "TDP" (stands for "Text-Document Processor"). You can run TDP either from MPE or from within ALIAS. The best introduction to HP editors is the introductory manual for the one called EDITOR.

PRIVELEGES: ALIAS is a secured system, meaning that ALIAS system administrators can limit the things that you can do. The list of things that you are allowed to do is referred to as your priveleges. It is important to realize that your actions can affect your priveleges to some extent. For example, if you create a scenario which uses some

data indirectly, you will not have data change priveleges for that data in the Data Base Updating system. Also realize that priveleges depend upon context: if you are running RELATE under your "R" user name you will not be allowed to change data base data, but if you are in the DBU you probably will be.

ON-LINE HELP: Most ALIAS software provides at least some on-line help. If you are uncertain of what you are supposed to do at any point, just give the "?" character as a command. You are likely to be presented with a menu offering a choice of many sorts of help (general description of what you should do, list of available commands, etc.); pick one or more kinds.

PARAMETER: Technical term for a variable or value which appears on a Command System parameter menu. Parameter settings typically influence how a given module operates. For example, setting a TIME UNITS parameter to "FISCYR" would be likely to produce different results than setting it to "MONTHS".

GATE: A particular kind of parameter, one which takes on the values of "ALL" or "LIST". By setting the value to "LIST" (even if that already appears to be its value) you will be presented with a page or list of items which you may turn on or off. See Section 4 for more information.

2.8 SUMMARY

A number of fairly abstract concepts have been presented in this Section. The most important ones to know before going on are:

1) CONTEXT: ALIAS is like a building with many rooms. You can do different things in different rooms. When you are finished with one task and want to perform another, you must "move" to a different context. There are about half a dozen different context types in ALIAS. By learning to recognize the types and the manner in which things are done in each one you will find yourself able to use ALIAS more effectively.

2) INSTRUCTIONS vs. PROCESSORS vs. DATA: Using ALIAS is the process of giving instructions to processors which will manipulate data. The data, however, is the most important part of the system. It is the heart of any study. Always remember that you have control of the data values through the Data Base Updating system and

the scenario system---you can change anything. The data forms a picture of the world: if you want to do an exercise in which a different world is envisioned (say more shipyards) you need only change the picture.

3) SCENARIOS: The ALIAS data base is divided into partitions which are like file drawers. Whenever you use ALIAS, you can only use one scenario at a time (though you may be able to "borrow" some data from other "drawers"). This ensures that you do not interfere with anyone else's work, and that they do not interfere with yours. However, you have some freedom to move data from one scenario to another, and you can always change which scenario you are working with.

4) DATA BASE: The ALIAS data base is a set of separate tables or files, each holding data about one well-defined subject. You can use the facilities of the RELATE Data Base Management System to sort, combine, extract, and summarize data from the table in ways that let you answer many different types of questions.

3.0 SAMPLE SESSION

This Section will guide you through a sample ALIAS session in which a number of the system's features will be demonstrated. Right-hand pages will show the session as it develops, while left hand pages will offer a running commentary on what is happening and why. A typical right hand page will show two screen "frames", separated by a line of "//////////" characters. The things typed by the user will be in **bold face**, while those printed by the computer will be in normal type.

The goal of the Section is to introduce you to the operation of ALIAS software by examples, not to thoroughly explain each feature and its options. Do not worry if there are steps whose rationale is unclear. If you come away with a general idea of how ALIAS works you will be prepared for the in-depth explanations of later sections.

The data you will see in the sample session is all notional data, not an accurate representation of the data NAVSEA uses. Also, the decisions made by the person running the session are not those of a seasoned NAVSEA analyst; if some of the decisions seem unrealistic to you, you are probably right!

The session does attempt to present in a realistic fashion how one might go about performing an acquisition program planning task with ALIAS, though. The task conducted in the session is design of a fiscal 1986 (Program Objective Memorandum) POM to support deployment of 17 carrier battle groups by the mid-1990s, and generation of a rough cost estimate for this POM.

If you have ALIAS user names and access to a terminal you can run the session as you read it. Be aware that you will probably not be able to duplicate it exactly, though, because the contents of the data base will have changed since this Guide was written. You should still be able to carry out most functions even if your results are somewhat different.

As the sample session develops you may begin to feel that ALIAS offers a truly bewildering array of options at each step. It is true that there are usually many different things you can do at any given point. This is the case because ALIAS tries to serve a large and diverse community of users, who require a large and diverse array of options. It is not necessary that you understand all the options in order to use the system, though. In many cases you can get along just fine without even knowing that many of them exist. The best way to learn how to use ALIAS is to concentrate on the options and capabilities which seem most relevant to your needs. In this way you will begin to get the feel of the system, which is the most important thing.

The facing page shows the ALIAS system map, which pictures all the menus currently offered by the Command System. It is a picture of the top-level options available.

When operating the system it is often helpful to think of your actions in terms of moving from one location to another on the map, or within one of the blocks on the map. This makes it easier and more natural to constantly keep track of your context (defined in Section 2), just as you keep track of where you are in the back of your mind when someone else is driving. Since your options depend on where you are in ALIAS, you can avoid much frustration by keeping track. If you become lost, you can turn back to the map and attempt to locate yourself. The map is available on-screen from any Command System menu by giving the command "?*".

*** ALIAS MENU SYSTEM MAP ***

```
                                       ..............
c = choice menu                        : TOP CHOICE :
p = parameter menu                     :    MENU    :
l = list menu                          :............:
                                             |
      |--------------|-------------|-------------|---------------|
.......c....... ......c..... .......c...... .....c...... ........c........
:     USER    : : OTHER HP : :  SCENARIO  : : ASSIGNER : : FORCE LEVELS :
: ENVIRONMENT : : PROGRAMS : :  CONTROL   : :          : :              :
:.............: :..........: :............: :..........: :..............:
      |                           ---------------|               |
.......p.......                   |                       .......p.......
: ENVIRONMENT :             ......p.......                : FORCE LEVEL :
: PARAMETERS  :             :  ASSIGNER  :                : PARAMETERS  :
:.............:             : PARAMETERS :                :.............:
                            :............:                       |
                  |---------------|-------------|         .......l.......
             .....l...... ......l...... ......l......     : REPAIR JOBS :
             : YARDS TO : : CLASSES TO : : JOB TYPES :    : CAUSING OUT-:
             :  ASSIGN  : :   ASSIGN   : : TO ASSIGN :    : OF FORCE    :
             :..........: :............: :...........:   :.............:
```

To use ALIAS you must log on to the SEA 90 account of the NAVSEA PMS 392 Hewlett-Packard 3000 computer. You will only be able to do this if you have been given user names and passwords by an ALIAS system administrator.

To log on, sit down at a terminal connected to the HP and press the RETURN key. Whenever you want to get HP's attention, or send a command off for processing, the RETURN key is the first one to try.

HP will respond with a ":" as a prompt. You should have two user names, one ending with an "A" and one ending with an "R". The "A" name is for running the ALIAS System Core and its attached modules; the "R" name is for running the RELATE Data Base Management System (DBMS). Type your "A" name in response to the colon. In the printed sample session the user is named "JOHN". He has user names "JOHNA" and "JOHNR".

HP will ask for your password. It will not appear on the screen as you type it in, for security reasons. Type it blind and hit the RETURN key (if you make a mistake, you have up to three tries to get it right; after that you must type in your user name again, but no harm is done).

A couple of bulletin-board like messages will appear, and you will get another colon prompt. Type "ALIAS". The ALIAS Core system will start up and will print a welcome message.

```
:HELLO JOHNA.SEA90
ENTER USER PASSWORD:

HP3000 / MPE IV C.B1.A2.  SAT, OCT 20, 1984,  11:20 AM
***************************************************************
WEEKLY BACK UP OF FILES IS NOW TAKING 8- 2400 FEET REELS
OF TAPE AND LASTING MORE THAN 2 1/2 HOURS.  IT IS IMPERATIVE
THAT USERS OF THE SYSTEM PURGE OLD FILES ON A REGULAR BASIS.
***************************************************************


Bulletin:
**********************************************************************
**********************************************************************
****************** DO NOT LEAVE YOUR TERMINAL ***********************
******************          UNATTENDED        ***********************
******************      SIGN    OFF   !!!!    ***********************
**********************************************************************
**********************************************************************
______________________

END OF PROGRAM

:ALIAS




                    ************************************
                    *                                  *
                    *          WELCOME TO ALIAS        *
                    *            VERSION 1.0           *
                    ************************************




                    > SYSTEM START-UP IN PROGRESS:
                     -Loading core system...
                     -Connecting to data base...
```

As explained in Section 2, ALIAS presents menus and screens. Both offer choices and accept commands, but screens are primarily fill-in-the-blank oriented while menus are tell-it-what-to-do-with-a-command oriented. The quickest way to tell one from the other is by the presence of inverse video---if there are bright white or green rectangles on the terminal, you're looking at a screen; if not, it's a menu.

The first thing you see whenever you run the ALIAS Core is a menu put up by the scenario system. The scenario system lets you choose what scenario (exercise/study) you want to work with. You must always have a "current scenario". Otherwise ALIAS will not know which data to retrieve from the data base to support your work.

Most ALIAS screens and menus print a few of the commands they will accept at the top of the display as a reminder. By looking at this help (top frame, facing page), JOHN sees he is supposed to give the name of the scenario he will work with initially. Alternatively, he can give one of the listed command characters. He chooses "" **to find out what scenarios are available.**

The response appears in the lower frame. Only two scenarios are currently available, and neither looks like a good one for an experimental session. JOHN concludes he had better make up a new scenario, which he is allowed to do at this point by giving the "+" command. By doing this JOHN is making sure that he does not interfere with anyone else's work by accidentally changing their data.

If you are following along, you can skip this step if there is a scenario available which you can use without destroying anyone's work.

```
                                                 Current scenario is

Scenario choice options include:
 ?      provides help                   |  ^    exits scenario choice system
 @      lists existing scenarios        |  +    moves you into scenario
 @name  lists the composition           |        creation menu
          of the 'name' scenario        |  &    re-displays this menu
 name   makes the 'name' scenario       |
        your current scenario           |
--------------------------------------------------------------------------------
                           SCENARIO CHOICE MENU
Name of scenario to use, or Command character: @
```

```
///////////////////////////////////////////////////////////////////////////////

                           CURRENT ALIAS SCENARIOS

SCENARIO        CREATOR   LAST USED    CREATED     READABLE BY  CHANGABLE BY
___________     _______   __________   _________   __________   __________
FIXIT           DBA       10/18/1984   4/17/1984   PUBLIC       PUBLIC
                COPY OF THE MAIN SCENARIO; TO BE USED FOR FIXING UP THE DATA
                FROM CSDS SO THAT SCENARIO MAIN IS AVAILABLE AS A BACKUP.
MAIN            DBA       10/18/1984   9/21/1983   PUBLIC       DBA
                THIS SCENARIO HOLDS THE MAIN DATA FOR THE ALIAS SYSTEM
                IT IS THE ONLY SCENARIO WHICH IS MAINTAINED BY SYSTEM DATA ENTRY

Hit <RETURN> for return to command menu:


Name of scenario to use, or Command character: +
```

Scenario creation is a multi-stage process. In the first stage, JOHN is prompted for some basic information about the scenario he wants to create: its name, whether other people will be allowed to use it and/or change its contents, and a description. The description can be several lines long.

Notice that JOHN could have given one of the command characters instead of the new scenario's name. In particular, if he had typed the "+" by accident in the previous menu, he could "back out" by giving the "^" command at this point. The "^" ("pop") command's effect is always to take you back to where you were before. When given in a Command System menu such as those listed on the system map shown a few pages ago, its effect will always be to pop you to a higher level on the map.

```
                                         Current scenario is
Scenario creation options include:
 ?      provides help                 |  ^     exits scenario choice system
 @      lists existing scenarios      |  &     re-displays this menu
 name   specifies the name of your    |  @name lists the composition
          new scenario                |          of the 'name' scenario
------------------------------------------------------------------------
                      SCENARIO CREATION—STAGE 1

NAME of new scenario, or COMMAND: demo

Do you wish to allow all ALIAS users to
examine the contents of this scenario? y

Do you wish to allow all ALIAS users to
modify the contents of this scenario? y

Please give a description of this scenario.  Terminate it with a '//' line.
.......................................................................
>scenario which goes with user manual sample session
>//
```

In stage two of scenario creation JOHN must specify the makeup or contents of his new scenario. A scenario can be thought of as a personal copy of the data in the data base. The copy can start as a clean slate, or as an actual copy of the data belonging to an existing scenario.

The data base is divided into groups or families along functional lines. Scenario makeup is specified by group, making it possible to take data from several existing scenarios. John, however, elects to have the new scenario (named "demo") start out as a copy of the "fixit" scenario. He responds "fixit" after the query for each data base group. (If he had wanted to start with a clean slate for any group, he would have responded "demo" to the appropriate "COMMAND or NAME:" prompt.)

The group names are short and rather mysterious. Here is what they stand for:

1) CURRJ: Current shipyard jobs, both new construction/conversion and repair. In Version 1.0 of ALIAS, the data involved is primarily schedules. "Current" jobs are those awarded but not yet delivered.

2) DESCJ: Job Descriptions. A job description tells how much time and/or resources it takes to do a shipyard job of a given type on a ship of a given class.

3) HISTJ: Historical shipyard jobs. Similar to the CURRJ group, but for jobs which have been completed.

4) MISCJ: Miscellaneous job-oriented information. Primary ship-class descriptions and individual ship retirement dates.

5) PARAMS: The name stands for "parameters", which in turn refers to the values which appear on some Command System menus. Parameters control the operation of many ALIAS modules. They are always "owned" by a particular scenario, so a group is required for their storage (note that the HP file group used is called "MNUREL", not "PARAMS").

6) PROJJ: Projected jobs. Like CURRJ and HISTJ, but shipyard jobs which have not yet been awarded.

7) YARDS: Information about shipyards.

Notice that JOHN is also asked if he will want to make changes to the data in each group. If he says yes, an actual copy of the source data he requests will be made for his use. If he says no, he will be using the data from the source scenario (fixit) "indirectly". More on this later.

```
                                                    Current scenario is

Scenario creation options include:
 ?      provides help                       |  ^      exits scenario choice system
 @      lists existing scenarios            |  &      re-displays this menu
 name   specifies the name of the           |  @name lists the composition
          scenario to take data from        |          of the 'name' scenario
          for a particular DB family.       |
```

SCENARIO CREATION—STAGE 2

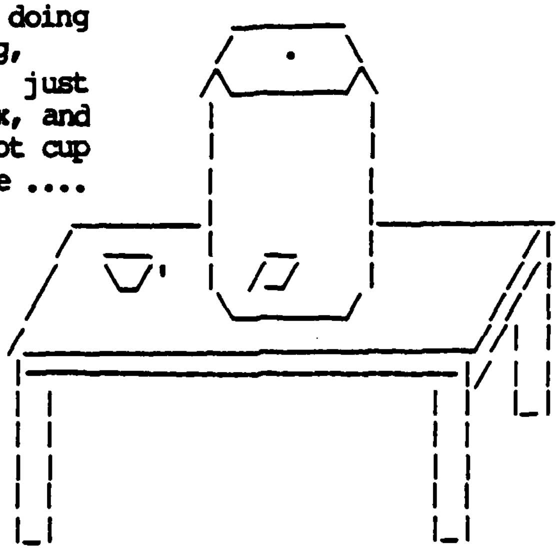
```
Please give the name of the scenario to take data for group CURRJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n

Please give the name of the scenario to take data for group DESCJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? y

Please give the name of the scenario to take data for group HISTJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n

Please give the name of the scenario to take data for group MISCJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n

Please give the name of the scenario to take data for group PARAMS   from.
COMMAND or NAME: fixit

Please give the name of the scenario to take data for group PROJJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? y

Please give the name of the scenario to take data for group YARDS    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n
```

After the questions have been answered for all the groups, ALIAS displays a series of messages. The first is the infamous "coffee cup", recommending that JOHN take a break. The coffee cup is a wry warning that things are about to get very slow. Some parts of ALIAS can take a long time to execute, particularly during peak daytime computer usage hours when everything is slow anyway. Scenario creation is one of these tasks; it can be counted on to take at least five minutes during the day and often much more.

The "Copying" messages report on progress of the creation. Note that the group name of each relation (the part in capitals after the "." in each message) corresponds to a group name on the previous page (except for ".MNUREL", which is where the "PARAMS" are stored). The only group names which appear are those for which JOHN responded "yes" when asked if he expected to change data in that group.

When the "LOADING PARAMETERS" message comes up ALIAS is almost ready for use.

```
Now, while I'm doing
some processing,
why don't you just
sit back, relax, and
have a nice, hot cup
of  C o f f e e ....
```



```
Copying data in relation NCJDAT.DESCJ
Copying data in relation NCJDCOM.DESCJ
Copying data in relation ASNPRM.MNUREL
Copying data in relation ENVRN.MNUREL
Copying data in relation FLREPT.MNUREL
Copying data in relation VALCLS.MNUREL
Copying data in relation VALYDS.MNUREL
Copying data in relation VLJTYP.MNUREL
Copying data in relation VLRJOB.MNUREL
Copying data in relation NCJOCOM.PROJJ
Copying data in relation NCJODAT.PROJJ
Copying data in relation REJODAT.PROJJ
Copying data in relation REJOCOM.PROJJ
```

```
************************************
*                                  *
*        LOADING PARAMETERS        *
*                                  *
************************************
```

When you run the ALIAS Core you always start out in the "Command System" context. A reminder of this fact is the "*ALIAS COMMAND SYSTEM*" label at the top of the menu. This label appears on every Command System menu.

The first menu to be presented by the Command System is always the "TOP" menu. In this menu six choices are available, corresponding to six broad types of activity.

Now that JOHN has created the scenario he will be working with he is ready to start designing his revised POM. He is operating on the assumption that the scenario he took his starting data from, FIXIT, was one containing a POM supporting a 15-carrier battle group Navy. He can therefore make a first guess at the new POM by just adding enough ship construction jobs to it to fill out two more battle groups.

The best tool for fast, high level design of an overall SCN POM is the Manual Assignment Editor ("assigner" for short). The assigner lets you create, inspect, and modify program designs using a timetable representation of the program schedules. This is much faster than working at the level of the individual schedules themselves. JOHN therefore chooses option 4.

ALIAS' response is to present another menu, this one with only two choices. JOHN can either look at the assigner initialization parameters, which are variables whose settings control what the assigner does, or he can go ahead and run the assigner. Since he doesn't know what parameter settings he inherited from the "fixit" scenario, he opts to look them over.

```
Menu is TOP               * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
                          TOP LEVEL ALIAS COMMAND MENU
   1.  CUSTOMIZE USER ENVIRONMENT
   2.  CALL NON-ALIAS PROCESSORS
   3.  DATA BASE UPDATING SYSTEM
   4.  MANUAL ASSIGNMENT EDITOR
   5.  FORCE LEVEL REPORT GENERATOR
   6.  SCENARIO CHOICE/MAKEUP SYSTEM

 COMMAND: 4
```

//

```
Menu is ASSIGN            * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
                        MANUAL ASSIGNER SPECIFICATIONS
   1.  ASSIGNER INITIALIZATION PARAMETERS
   2.  EXECUTE THE ASSIGNER

 COMMAND: 1
```

The parameters are set to display assignments in terms of fiscal year of ship award (1 and 7) from 1980 through 2000 (2 and 3). Only projected jobs will be displayed (10). When the assigner creates new schedules based on JOHN's changes to the POM, it will attempt to generate them such that the start-construction milestones of the ships in each program at each yard will be spread evenly over time (8 and 9). On the assigner display screen, shipyards will be displayed in the order they were shown during the last assigner run, while ship classes will be alphabetized within yards (11 and 12). The assigner will re-type its displays only when JOHN tells it to, not automatically after each command (13). Parameters 4, 5, and 6 are "gates" to "list menus". Their pages offer lists of the yards, ship classes, and job types that the assigner is capable of working with. By marking the members of the lists as "on" or "off", JOHN can indicate which yards/classes/job types he wants to work with for this run. A setting of LIST on the parameter menu (as opposed to ALL) indicates that some of the list members <u>may</u> be turned off.

If you are running a sample session as you read this, you can look at one of the menus by giving the command "4=LIST".

JOHN doesn't worry about the list statuses, though. He sets the time period of interest to fiscal 1986-94 and the shipyard display order to alphabetic, and leaves the menu.

Back in the assigner choice menu again, he asks that the assigner be executed. The assigner informs him of its progress as it reads the existing construction schedules for projected jobs in the DEMO scenario.

```
Menu is ASNPRM              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
               MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                 = FISCYR       (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE             =  1/ 1/1980   (MM/DD/YYYY)
 3.  ENDING DATE               = 12/31/1999   (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS      = LIST         (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES    = LIST         (ALL/LIST)
 6.  CANDIDATE JOB TYPES       = LIST         (ALL/LIST)
 7.  DISPLAY BASIS             = AWD          (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS              = START        (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE               = PROGRAM      (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION         = PROJ         (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER      = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER       = INPUT ORDER  (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH              = OFF          (ON,OFF)

   COMMAND: 2=10/1/1985
   COMMAND: 3=9/31/1994
   COMMAND: 12=ALPHA
   COMMAND: ^

Saving parameter settings...

/////////////////////////////////////////////////////////////////////////

Menu is ASSIGN              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
                      MANUAL ASSIGNER SPECIFICATIONS
      1.  ASSIGNER INITIALIZATION PARAMETERS
      2.  EXECUTE THE ASSIGNER

   COMMAND: 2

Starting up Manual Ship Assigner.  Expect a one to five minute delay.
Loading assignments for yard AVONDALE
Loading assignments for yard BIW
Loading assignments for yard EB GROT
Loading assignments for yard GDQ
Loading assignments for yard INGALLS
Loading assignments for yard MARINET
Loading assignments for yard NASSCO
Loading assignments for yard NNEWS
Loading assignments for yard PENNSHIP
Loading assignments for yard PETERSON
Loading assignments for yard PH NSY
Loading assignments for yard TACOMA
Loading assignments for yard TODD SEA
Loading assignments for yard UNKBLDR
```

Now we are in a new context, the assignments editor context. Oriented towards data display and modification, the assigner's displays are pages rather than menus or screens---notice that there is no inverse video, if you are running a session, and no list of choices. As in the parameter menu he just worked with, JOHN will make changes by using editor-like commands (Add, Delete, Modify, etc.) rather than by filling in or changing fields on the display.

There is a lot of information packed onto an assigner display. Rather than explain it all at once, we will discuss it a bit at a time as the session continues. Notice for now that the top line tells JOHN that he is in a new context ("*SHIP ASSIGNMENTS*" rather than "* ALIAS COMMAND SYSTEM *"), that the period numbers forming column headers represent fiscal years (FISCYR), and that he is using scenario DEMO.

The body of the assigner table is marked out in subsections, one per yard, by grid lines. Each individual ship class that a yard is to do work on has its own line within the subsection; the numbers on the line tell how many awards the yard will receive each fiscal year for jobs in that class (they are awards because that was the display basis date setting in the assigner's parameter menu---they could just as easily be starts or deliveries). For example, AVONDALE will get contracts to build two LSD-41's in fiscal 1986 and one in fiscal 1987.

Notice that each yard has a number, which follows its name, and each class has a number preceding its name. These numbers are the means of picking out a line of the display to make changes to. JOHN wants to give BATH two CG-47's in fiscal '89 instead of just one; to do this he asks to modify the assignments for yard 2, class 1 by giving the "M 2.1" command. The assigner responds with a miniature version of the display showing the current assignments for 2.1, the bottom line of which is a row of dots which act as a typing guide. All JOHN has to do is space over and put a "2" in the column he wants to alter. BATH's CG-47 assignments in other years will remain unchanged.

That was the only change JOHN wanted to make to the assignments showing on this page of the assigner display. He tells the assigner to go on to the next page with the "+" command. Since he had left AUTO REFRESH set to "off" in the parameters menu, the assigner doesn't actually type out page 2---it just gets ready to. JOHN asks it to show the new page with the "&" command

```
Scenario: DEMO                *SHIP ASSIGNMENTS*    Page  1A    Time in: FISCYR
Yard      Period:| 1  2  3  4  5  6  7  8  9                              | 9
    Shipclass T|86 87 88 89 90 91 92 93 94                              |TOT
AVONDALE     #01|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 LSD-41      | 2  1                                                  | 3
  2 LSD-49      |       L2  2  2                                        | 6
  3 T-AO-187    | 2  2  2  2  2                                         | 10
BIW          #02|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 CG-47       | 1  1  1  1                                            | 4
  2 DDG-51      |    Y1 F2  1  2                                        | 6
EB GROT      #03|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 SSBN-726    | 1  1  1  1  1                                         | 5
  2 SSN-21      |          L1                                           | 1
  3 SSN-688     | 2  2  2  1  2                                         | 9
GDQ          #04|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 AE          |       Y1  1  1                                        | 3
  2 AG          |L1                                                     | 1
  3 AO-187     c|       1  1                                            | 2
                |--+--+--+--+--+--+--+--+--|---------------------------|---
 14  33   TOTALS|29 24 33 30 27     1                                   |144
(?=help) > M 2.1

Modify assignment to yard BIW
(Blanks denote no change; use zero for assignment deletion)
          Period:| 1  2  3  4  5  6  7  8  9
    Shipclass T|86 87 88 89 90 91 92 93 94
  1 CG-47       | 1  1  1  1
                .. .. .. .. .. .. .. .. ..
                          2

(?=help) > +

(?=help) > &
```

Notice that the page number at the top right now reads "2A" rather than "1A". Although it will not be used in this session, the assigner can page right and left as well as up and down. It can work with an effectively unlimited number of vertical or up/down pages, and with up to 13 horizontal or right/left pages. The 13 pages can hold up to 20 columns each, giving the assigner a 260-period capacity. You can make a 260-year projection! Or more usefully, you can look at a five-year program in terms of months, a relatively high level of time-resolution.

In the page number at the top of the display, the number tells what vertical page JOHN is on, while the letter tells what horizontal page.

Flip back to the previous page and compare the TOTALS line at the bottom with the one on the facing page. Notice that the "89" column is now greater by one (31 rather than 30, since an extra CG-47 was assigned), and the grand total at the bottom right is also (145). These totals are not single-page totals, but are for all pages.

On this page JOHN makes a modification to CG-47 assignments at INGALLS in a fashion similar to the BATH change. He then wants to add assignments to the NNEWS yard for a class NNEWS does not yet have. He asks to Add to yard 8 ("A 8"), and is given a prompt similar to the one used for modify, except that he must fill in the ship class name as well as the assignments. Then he asks for the next page using the "+" and "&" commands.

```
Scenario: DEMO                  *SHIP ASSIGNMENTS*   Page  2A   Time in: FISCYR
Yard     Period:| 1  2  3  4  5  6  7  8  9                          |  9
   Shipclass T|86 87 88 89 90 91 92 93 94                          |TOT
INGALLS      #05|--+--+--+--+--+--+--+--+--|------------------------------|---
 1 BB-61      r|    1                                                |  1
 2 CG-47       | 2  2  2  1                                          |  7
 3 DDG-51      |   F1  2  2  2                                       |  7
 4 LHD         |      L2  1  1                                       |  4
MARINET      #06|--+--+--+--+--+--+--+--+--|------------------------------|---
 1 MCM-1       | 2  1                                                |  3
 2 MSH-1       |F4  4  4  4                                          | 16
NASSCO       #07|--+--+--+--+--+--+--+--+--|------------------------------|---
 1 AO-187     c|          1  2                                       |  3
 2 AOE         |   L1 F1  1  1                                       |  4
 3 AR          |            L1                                       |  1
 4 LPD-4      c|       1  1  2                                       |  4
NNEWS        #08|--+--+--+--+--+--+--+--+--|------------------------------|---
 1 MTS        c|    1                                                |  1
 2 SSN-688     | 2  2  2  1  2                                       |  9
                |--+--+--+--+--+--+--+--+--|------------------------------|---
 14  33  TOTALS|29 24 33 31 27     1                                 |145

(?=help) > M 5.2

Modify assignment to yard INGALLS
(Blanks denote no change; use zero for assignment deletion)
         Period:| 1  2  3  4  5  6  7  8  9
     Shipclass T|86 87 88 89 90 91 92 93 94
  2 CG-47       | 2  2  2  1
                 .. .. .. .. .. .. .. .. ..
                          2

(?=help) > A 8

Make new assignment for yard # 8, NNEWS
         Period:| 1  2  3  4  5  6  7  8  9
     Shipclass T|86 87 88 89 90 91 92 93 94
     ........... .. .. .. .. .. .. .. .. ..
     CVN-68        1        1

(?=help) > +

(?=help) > &
```

On the new page JOHN needs to add a yard that doesn't appear at all, TODD LA. A plain Add ("A") command with no numbers after it initiates this process. JOHN is first prompted for the name of the yard, and then for the first assignment in the yard in the usual fashion.

Notice on this page that several of the ship classes have lower-case characters following their names, like "c" after MTS in the NNEWS yard. These characters indicate that the work being done on these classes is something other than new construction (a blank, the default, indicates new construction). The "c" stands for conversion, "s" for SLEP, and "r" for reactivation. It is important to know that the assigner is capable of generating schedules only for what are called "new construction-type jobs", which at present are limited to new construction, conversions, and reactivations. It can read "repair-type job" schedules from the repair job data base and display them in the assigner format, but any changes made to, e.g., SLEP assignment lines will not be saved.

JOHN asks that the current page be re-displayed by giving the "&" command.

```
Scenario: DEMO                  *SHIP ASSIGNMENTS*   Page  3A   Time in: FISCYR
Yard     Period:| 1  2  3  4  5  6  7  8  9                                 |  9
    Shipclass T|86 87 88 89 90 91 92 93 94                                  |TOT
NNEWS        #08|--+--+--+--+--+--+--+--+--|--------------------------------|---
  1 CVN-68      | 1     1                                                   |  2
  2 MTS        c|    1                                                      |  1
  3 SSN-688     | 2  2  2  1  2                                             |  9
PENNSHIP     #09|--+--+--+--+--+--+--+--+--|--------------------------------|---
  1 LPD-4      c|          2  1                                             |  3
  2 T-ACS      c| 3  2  2                                                   |  7
  3 T-AVB      c| 1                                                         |  1
PETERSON     #10|--+--+--+--+--+--+--+--+--|--------------------------------|---
  1 MCM-1       | 2                                                         |  2
PH NSY       #11|--+--+--+--+--+--+--+--+--|--------------------------------|---
  1 CV-63      s| 1                 1                                       |  2
TACOMA       #12|--+--+--+--+--+--+--+--+--|--------------------------------|---
  1 T-AGOS-1    | 3                                                         |  3
TODD SEA     #13|--+--+--+--+--+--+--+--+--|--------------------------------|---
  1 AFDM        |       1     1                                             |  2
      (more)    |--+--+--+--+--+--+--+--+--|--------------------------------|---
 14  34   TOTALS|30 24 33 34 27       1                                     |149

(?=help) > A

Give yard name please (Type "^" to cancel) > TODD LA

Make new assignment for yard #13, TODD LA
         Period:| 1  2  3  4  5  6  7  8  9
    Shipclass T|86 87 88 89 90 91 92 93 94
    .......... .. .. .. .. .. .. .. .. ..
    DDG-51          F1  2  2  2

(?=help) > &
```

JOHN's strategy in making these changes has been to add assignments only in those classes which he guesses to be the binding constraint on formation of new carrier battle groups. He is assuming that the FIXIT scenario, from which he took the POM he has been building on, was a 15-carrier battle group scenario, and that a battle group requires 1 carrier, 1 cruiser, and 4 DDGs. These assumptions are only as correct as JOHN's knowledge of the FIXIT scenario, which includes records of historical and current jobs as well as the projected jobs he has just inspected as changed.

This dependency of an ALIAS analysis on a broad range of data base data, a fact which results from the fundamental dependency of ship acquisition program analysis on lots of data, is one of the most difficult features of ALIAS to get used to. If part of the data in a scenario you are working with is in error, or is merely not what you think it is, your results are likely to be affected. This is why JOHN created a new scenario for this sample session rather than just using FIXIT---the owner of FIXIT would have returned to find his data changed, perhaps in subtle ways that would not show up for a while. It is thus important that all ALIAS users exercise restraint in making changes to scenarios not their own.

JOHN inspects the assigner page and is satisfied. He gives the Quit ("Q") command, which triggers the assigner's schedule creation and update pass.

```
Scenario: DEMO                    *SHIP ASSIGNMENTS*   Page  4A   Time in: FISCYR
Yard       Period:| 1  2  3  4  5  6  7  8  9                         |  9
    Shipclass    T|86 87 88 89 90 91 92 93 94                         |TOT
PENNSHIP       #09|--+--+--+--+--+--+--+--+--|------------------------|---
  3 T-AVB        c| 1                                                 |  1
PETERSON       #10|--+--+--+--+--+--+--+--+--|------------------------|---
  1 MCM-1         | 2                                                 |  2
PH NSY         #11|--+--+--+--+--+--+--+--+--|------------------------|---
  1 CV-63        s| 1                 1                               |  2
TACOMA         #12|--+--+--+--+--+--+--+--+--|------------------------|---
  1 T-AGOS-1      | 3                                                 |  3
TODD LA        #13|--+--+--+--+--+--+--+--+--|------------------------|---
  1 DDG-51        |   F1  2  2  2                                     |  7
TODD SEA       #14|--+--+--+--+--+--+--+--+--|------------------------|---
  1 AFDM          |       1     1                                     |  2
  2 DDG-51        |      Y1 F2  1                                     |  4
UNKBLDR        #15|--+--+--+--+--+--+--+--+--|------------------------|---
  1 LPDS-?       s|       2  1  2                                     |  5
  2 LPDSW-?      s|    1  1  2  1                                     |  5
                  |--+--+--+--+--+--+--+--+--|------------------------|---
 15  35     TOTALS|30 25 35 36 29     1                               |156
(?=help) > Q
```


61

The process of creating complete schedules from the assignments and storing them in the data base is rather time consuming, so the assigner asks JOHN if he wants to skip it. He might want to skip it if he had run the assigner only to inspect the POM, and had made no changes. Since he wants his changes implemented, though, he cannot skip it this time.

The assigner lists the yards as it processes their assignments to give an idea of its progress. Notice that it issues a warning message in the middle of listing the yards, to the effect that it cannot find a job description for the lead LSD-49 at Avondale. It says it is using the job description for construction of an "ORDFOL" (ordinary follow) LSD-49 instead. If you look back a few pages at the assignments in the Avondale yard you will see that an "L" appeared in the fiscal 1988 column of LSD-49 assignments, indicating that one of the two awards in that year will be for a lead ship.

ALIAS differentiates among jobs by this concept of "series type". When the assigner makes up a schedule for a lead ship, it expects to find a job description for lead ships of that class. If it cannot find one, it uses the ordinary follow job description instead, issuing a warning so the user knows what's happening.

There are a couple of reasons why this message might be appearing for JOHN. One is that there may be no difference between lead and follow LSD-49 construction jobs; in such a case, the owner of the FIXIT scenario (where JOHN got his POM) may not have bothered to put in a lead job description. Alternatively, the FIXIT owner may have put in the lead LSD-49 schedule manually using the Data Base Updating system, and may have forgotten to enter a job description for it. Any schedule can be entered manually---job descriptions are required only by the assigner.

In any case, JOHN makes a mental note of the message, resolving to check out the LSD-49 job descriptions.

When schedule creation completes JOHN is returned to the menu he called the assigner from. He enters the "/" command, the meaning of which is "pop to the topmost menu".

The assigner will now update ship schedules for the current scenario.

Only projected ship schedules will be updated. Any changes made during this session which imply changes to historical or current job schedules will be ignored.

If you made NO CHANGES, or if you want your changes DISCARDED, considerable time can be saved by skipping this update. If you do skip it, any changes you have made will be lost.

Do you want to skip the update? N

```
Opening data base files for update.
Updating projected ships data base.
Updating schedules for yard AVONDALE

UNABLE TO FIND JOB DESCRIPTION INFO FOR CLASS  LSD-49
AND SERIES TYPE LEAD   USING SERIES TYPE       ORDFOL INSTEAD
Updating schedules for yard BIW
Updating schedules for yard EB GROT
Updating schedules for yard GDQ
Updating schedules for yard INGALLS
Updating schedules for yard MARINET
Updating schedules for yard NASSCO
Updating schedules for yard NNEWS
Updating schedules for yard PENNSHIP
Updating schedules for yard PETERSON
Updating schedules for yard PH NSY
Updating schedules for yard TACOMA
Updating schedules for yard TODD LA
Updating schedules for yard TODD SEA
Updating schedules for yard UNKBLDR
Updating hull numbers in schedules.
Update Complete. Display buffers purged.  Closing DB.
```

//

```
Menu is ASSIGN          * ALIAS COMMAND SYSTEM *   Scenario is DEMO
--------------------------------------------------------------------------
                    MANUAL ASSIGNER SPECIFICATIONS
   1.  ASSIGNER INITIALIZATION PARAMETERS
   2.  EXECUTE THE ASSIGNER

 COMMAND: /
```

In the Command System the topmost menu is the one first displayed when the ALIAS Core started up. Getting to this menu is a common act, since it is the starting point for any action---that's why there is a special command for doing so.

Having made his rough-cut changes to the POM, JOHN is ready to see if 17 carrier battle groups will result. The Force Level Report Generator is the tool for finding this out, so he chooses option 5.

Oops! He's forgotten something, and must go back to the top menu again.

```
Menu is TOP                * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
                         TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 5
```

//

```
Menu is FLRPTG             * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
                           FORCE LEVEL GENERATOR
    1.  FORCE LEVEL REPORT INITIALIZATION PARAMETERS
    2.  EXECUTE FORCE REPORT GENERATOR
    3.  PREPARE BATTLE GROUP REPORT

  COMMAND: /
```

He has to designate which printer he wants the report to come out on. The printer designation is part of his "user environment", so he chooses the "Customize User Environment" and then the "Environment Parameters" options.

```
Menu is TOP                 * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
                         TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 1
```

//

```
Menu is ENVIRC              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
                            ENVIRONMENT CONTROL
    1.  ENVIRONMENT PARAMETERS
    2.  SET LPRNTS (DEBUG SWITCHES)

  COMMAND: 1
```

There are three things that can be configured by setting Environment parameters. The type of terminal you are using is usually detected automatically by ALIAS, but if it guesses wrong you can override its guess by setting parameter 1 to the proper type. If ALIAS has your terminal type wrong it will be unable to clear the display or manage fill-in-the-blank screens for you.

When ALIAS prints a report, it generally sends it to the printer specified by the "DEVICE TO PRINT TO" parameter. DAISY is the SEA 90 daisy wheel printer, PRINTER the fast SEA 90 dot matrix printer, and LP the PMS 392 line printer. JOHN wants his output on the fast dot matrix, so he sets option 2 to PRINTER (notice he only has to specify the first few letters; the Command System figures out what he means).

Once he has made this setting ALIAS will remember it. He will not need to come back to the Environment Parameters menu until he changes scenarios (parameters are always a part of a scenario, so if you work with a different one they may be different) or until he wants to make a change.

The "PRINT COMMANDS ON MENUS?" parameter is a good one to set to "YES" if you are a new ALIAS user. Remember how the first few menus in this sample session, the ones where the scenario was being chosen/created, had a list of the most-used commands printed at the top? Such a list will appear at the top of all Command System menus if this parameter is set to "YES". Some users set it to "NO" so that the menus are typed faster on the screen.

Having made his setting, JOHN pops back to the top menu and then picks the force report option again.

```
Menu is ENVRN              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
                        USER ENVIRONMENT PARAMETERS
 1.  TERMINAL TYPE           = PC          (HP2623,SBRAIN,HZ15,HZ14,HP2647,PC
 2.  DEVICE TO PRINT TO      = TERMINAL    (TERMINAL,DAISY,LP,PRINTER)
 3.  PRINT COMMANDS ON MENUS? = NO         (YES,NO)

  COMMAND: 2=PR

  COMMAND: /

Saving parameter settings...
```

```
////////////////////////////////////////////////////////////////////////////
```

```
Menu is TOP                * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
                        TOP LEVEL ALIAS COMMAND MENU
   1.  CUSTOMIZE USER ENVIRONMENT
   2.  CALL NON-ALIAS PROCESSORS
   3.  DATA BASE UPDATING SYSTEM
   4.  MANUAL ASSIGNMENT EDITOR
   5.  FORCE LEVEL REPORT GENERATOR
   6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 5
```

In the Force Level choice menu, JOHN decides he had better look at the parameters which will control the report generators' operation, since the values he inherited from FIXIT may not be to his liking. He chooses option 1 to bring up the parameter menu.

When KEEP REPORT ON-LINE is set to YES, the report generators will write their output to a disk file as well as to the designated printer. This makes it possible to edit the report output.

The REPORT START DATE and REPORT END DATE parameters determine the period of time that reports should be generated for. In combination with TIME PERIOD LENGTH, they determine the number of periods/columns that will appear on the output. Note that no force report may have more than 20 periods, since that is the most that can fit on a 132 column printer page.

The RETIRE SHIPS BY parameter controls how ship retirement dates are computed. If DATE, the report generators will use any retirement dates they find in the data base, and a table of standard service lives for ships with no retirement date specified. If LIFE, the table of standard service lives is used even for ships with specified retirement dates, as long as the dates are <u>after</u> the date the report is being run (i.e. ships already out of the force cannot be un-retired by this setting). The purpose of the LIFE option is to allow studies of service life variations to be conducted without having to change those retirement dates which are specified.

Each ship will retire within a given period, say within a calendar year. The IN FORCE DAY parameter specifies the rule by which it is decided if the ship is counted in the force that period. If END it is not, if BEGIN it is.

PROGRAM MILESTONE is complicated and not of interest during this sample session, so discussion of it will be postponed. OUT OF FORCE REPAIR JOBS is a gate to a list of job type codes for repair jobs which cause a ship to be temporarily out of the force.

JOHN decides he wants the report to cover 1986-1999, and that he wants a look at the list represented by parameter 8. The way to bring up the list is to re-set its value to "LIST" (8=L is a shorthand command for 8=LIST).

```
Menu is FLRPTG            * ALIAS COMMAND SYSTEM *   Scenario is DEMO
-----------------------------------------------------------------------
                        FORCE LEVEL GENERATOR
   1.  FORCE LEVEL REPORT INITIALIZATION PARAMETERS
   2.  EXECUTE FORCE REPORT GENERATOR
   3.  PREPARE BATTLE GROUP REPORT

  COMMAND: 1
```

//

```
Menu is FLREPT            * ALIAS COMMAND SYSTEM *   Scenario is DEMO
-----------------------------------------------------------------------
        FORCE LEVEL AND BATTLEGROUP REPORT GENERATOR PARAMETERS
 1.  KEEP REPORT ON-LINE         = YES          (YES,NO)
 2.  REPORT START DATE           =  1/ 1/1986   (MM/DD/YYYY)
 3.  REPORT END DATE             = 12/31/2005   (MM/DD/YYYY)
 4.  RETIRE SHIPS BY             = LIFE         (LIFE,DATE)
 5.  TIME PERIOD LENGTH          = CALYR        (DAY,WEEK,MONTH,QTR,CALYR)
 6.  IN FORCE DAY                = END          (BEGIN,END)
 7.  PROGRAM MILESTONE           = APPROP       (APPROP,AWD,DELIV)
 8.  OUT OF FORCE REPAIR JOBS    = LIST         (ALL/LIST)

  COMMAND: 3=12/31/1999

  COMMAND: 8=L
```

There are only three repair job type codes listed as part of the DEMO scenario, and all of them are "off" the list. "On" list members have asterisks next to their names, and there are no asterisks. JOHN decides he wants any ships undergoing SLEPs to not be counted as deployable, so he must turn on the list member named SLEP. To do so, he need only enter its number, 3 (if it was on and he wanted it off, he would also enter "3"; giving a list member's number is like flipping a toggle switch).

After making the change, JOHN asks to go back to the parameter menu; when he gets there, he asks to go back to the Force Level choice menu.

```
Menu is FLREPT              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
          REPAIR JOBS THAT REMOVE A SHIP FROM FORCE DURING EXECUTION
--------------------------------------------------------------------------
   1.   REFUEL                           3.   SLEP
   2.   REPAIR
--------------------------------------------------------------------------

 COMMAND: 3

 COMMAND: ^

Saving list on/off settings...
```

//

```
Menu is FLREPT              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
           FORCE LEVEL AND BATTLEGROUP REPORT GENERATOR PARAMETERS
 1.  KEEP REPORT ON-LINE      = YES          (YES,NO)
 2.  REPORT START DATE        =  1/ 1/1986   (MM/DD/YYYY)
 3.  REPORT END DATE          = 12/31/1999   (MM/DD/YYYY)
 4.  RETIRE SHIPS BY          = LIFE         (LIFE,DATE)
 5.  TIME PERIOD LENGTH       = CALYR        (DAY,WEEK,MONTH,QTR,CALYR)
 6.  IN FORCE DAY             = END          (BEGIN,END)
 7.  PROGRAM MILESTONE        = APPROP       (APPROP,AWD,DELIV)
 8.  OUT OF FORCE REPAIR JOBS = LIST         (ALL/LIST)

 COMMAND: ^

Saving parameter settings...
```

He is ready to run the report now. He wants a battle group report, which will give force levels in terms of numbers of deployable battle groups. The alternative is a "raw" force level report giving the number of ships of each type available in each period. To get the battle group report, he chooses option 3.

Menu is FLRPTG * ALIAS COMMAND SYSTEM * Scenario is DEMO

FORCE LEVEL GENERATOR

1. FORCE LEVEL REPORT INITIALIZATION PARAMETERS
2. EXECUTE FORCE REPORT GENERATOR
3. PREPARE BATTLE GROUP REPORT

COMMAND: 3

//

The first thing the battle group report generator does is ask for the name of a "FORMAT CONTROL FILE". The force level report generators are unusual among ALIAS modules in that they require more input than parameters, lists, and data base values. The format control file is where you specify exactly what you want on the report, and how you want it placed on the pages. We will look at such a file in detail later in the session. For now, JOHN has decided to use the one called BGPOM86 in the .FMTFIL group.

The report generator then complains that the file FLREPT exists in the log-on group, i.e. in the file directory which holds JOHN's personal files. JOHN had left the KEEP REPORT ON-LINE parameter set to YES, i.e. he had specified that he wanted the text of the report saved in a file as well as printed. The report generators automatically save to a file called FLREPT, unless they find that a file of that name (probably holding a report run earlier) already exists. In that case they ask for an alternate file name. JOHN gives BGDEMO as the name of the file to save into.

The report generator then constructs the report by searching the data base for ships active in the period specified. The data base search is typically very time-consuming, so the report generator types out the names of the classes it is looking for as a means of reporting its progress.

```
Starting up Battle Group Report Generator. Please stand by.
NAME OF FORMAT CONTROL FILE (name.group, or ^) BGPOM86.FMTFIL

The file FLREPT already exists in your group.

The force level modules usually store reports in that
file when you request that the report be saved on disk.

Since that file is in use, what file would you like to store
the report in? (name <= 8 letters, or / to not save report): BGDEMO

Opening required data base files...
Looking for ships in data base...
Looking for class BB-61
Looking for class CG-16
Looking for class CG-26
Looking for class CG-47
Looking for class CGN-25
Looking for class CGN-35
Looking for class CGN-36
Looking for class CGN-38
Looking for class CGN-9
Looking for class CV-41
Looking for class CV-59
Looking for class CV-63
Looking for class CV-67
Looking for class CVN-65
Looking for class CVN-68
Looking for class DD-963
Looking for class DDG-2
Looking for class DDG-37
Looking for class DDG-51
Looking for class DDG-993
Looking for class FF-1040
Looking for class FF-1052
Looking for class FFG-1
Looking for class FFG-7
```

A copy of the output which resulted is shown in the top frame opposite. Five types of "battle group" are reported on here: carrier battle groups, surface action groups, marine amphibious forces, supply ships escorts, and convoys. The number of each group deployable in each year is given. Ships of each type that were "left over" after all the groups were made up (i.e. which were a member of no group) are totalled by type at the bottom.

JOHN immediately sees he was mistaken about the nature of the scenario he started out with. Not only does he not achieve 17 carrier battle groups in the 90's, the number achieved all along is well below what he was expecting. And yet there are plenty of aircraft carriers, in fact 17 are available from 1997 on (add up the CARRIER BG row and the CARRIER balance row). The problem must be with availability of other ships which make up a battle group. JOHN immediately decides that DDG availability must be the binding constraint, since there is a 0 balance for that ship type over the whole period.

Why is this availability unrealistically low? When you are faced with such a problem, the best thing to do is go ask the person whose scenario you used as a starting point. But for the purposes of this sample session, it will be more interesting if we have JOHN play detective instead.

There are only two reasons why this can be happening, he reasons. One might be that somehow the construction job schedules for a number of currently active DDG's never were entered in the data base for his scenario (the force level report generators look for those schedules to determine what ships enter the force when). The other reason might be that they are being retired too early.

To find out, JOHN will have to look at the contents of the data base. Since he is interested in a fairly significant volume of data (construction schedules for several DDGs), the best way to look is with the RELATE Data Base Management System's query language. To use RELATE, JOHN will have to log on as JOHNR.

When the battle group report generator finished it returned JOHN to the force report choice menu. He tells ALIAS he wants to finish his session via the "Q" command.

DEPLOYABLE BATTLE GROUP PROJECTION FOR POM-86
BASED ON SURFACE COMBATANT REQUIREMENTS ONLY
(ALL DATA NOTIONAL)

CALENDAR YEAR	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
BATTLEGROUP														
CARRIER BG	9	9	9	9	7	5	5	5	6	7	7	7	7	7
SURFACE AG														
MARINE AF					1	1		1			1	1	1	1
SUPPLY ESCORT	1	1	1	1	1	1	1	1		1				
CONVOY	10	10	10	10	8	10	10	10	10	10	9	9	9	9
BALANCE														
CARRIER	4	4	4	5	7	10	9	10	10	9	9	10	10	10
BB			1	1	1	1	1	1	1	1	1	1	1	1
CRUISER	7	7	7	7	7	9	9	7	8	7	5	3	3	3
DDG														
DD	2	2	2	2		2	10	2	9	6				
FFG														
FF	33	33	34	34	40	40	50	40	47	37	31	26	21	16

```
////////////////////////////////////////////////////////////////////////////////

Menu is FLRPTG              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------------
                            FORCE LEVEL GENERATOR
    1.  FORCE LEVEL REPORT INITIALIZATION PARAMETERS
    2.  EXECUTE FORCE REPORT GENERATOR
    3.  PREPARE BATTLE GROUP REPORT

  COMMAND: Q

Are you sure you want to terminate your ALIAS session? Y

END OF PROGRAM
```

When he receives the ":" prompt after the ALIAS Core program ends, he logs on as JOHNR using the same procedure as during his initial log-in. Then he runs RELATE, the DBMS. His first RELATE command (OPEN FILE) is a request to access the relation in which historical construction schedules are stored. To test the first hypothesis about the "missing DDGs" (that their construction jobs don't appear in this relation for the DEMO scenario), he gives a SELECT command designed to return one record per DDG construction job in ncjodat.histj. To get only one record per job, he asks that they be returned UNIQUE BY SCENARIO,CLASS,HULL, COMNUM (COMNUM is "commissioning number"). To get only ships actually built, he asks that DATETYPE="ACTUAL". To get only DDGs, he asks that the class name have the string "DDG" in it somewhere (MATCH(CLASS,"DDG")<>0). To get the information for the DEMO scenario only, he asks for SCENARIO="FIXIT".

Why "FIXIT" and not "DEMO"? Because when JOHN made up the DEMO scenario, he answered "NO" to the question "Will you want to change data in the .HISTJ group?" By giving that answer, and FIXIT as the name of the source scenario, he was implicitly asking to use FIXIT's data "indirectly" during ALIAS runs. If he had answered "YES", then a copy of the .HISTJ data for FIXIT would have been made, with scenario field values of DEMO.

So to find out what data is in ncjodat.histj for scenario DEMO, JOHN actually has to ask what is there for scenario FIXIT in this case. This is a subtle point that it's easy to forget about. Later in the sample session there will be an example of how to find out about this "composition" aspect of an already existing scenario.

You might be wondering why anyone would ever not want their own copy of the data for all groups, given this possibility for confusion. The answer is that if you have your own copy, then you are responsible for all the data updating necessary to keep it current! That can be a big job, especially for the .CURRJ group, so it is almost always best to use indirectly the data for a scenario that you know is receiving data updating service. Indirect use also saves disk space.

```
:HELLO JOHNR.SEA90
ENTER USER PASSWORD:

HP3000 / MPE IV C.B1.A2.  SAT, OCT 20, 1984,  12:16 PM
*************************************************************
WEEKLY BACK UP OF FILES IS NOW TAKING 8- 2400 FEET REELS
OF TAPE AND LASTING MORE THAN 2 1/2 HOURS.  IT IS IMPERATIVE
THAT USERS OF THE SYSTEM PURGE OLD FILES ON A REGULAR BASIS.
*************************************************************


Bulletin:
*******************************************************************
*******************************************************************
***************** DO NOT LEAVE YOUR TERMINAL **********************
*****************         UNATTENDED         **********************
*****************      SIGN    OFF   !!!!    **********************
*******************************************************************
*******************************************************************
```

```
END OF PROGRAM

:RELATE
Comment  MAKE SURE HP2934A IS TURNED ON BEFORE PRinting,
Comment  OTHERWISE YOU'LL LOSE SPOOLER OWNERSHIP.
FILE rdbecat.pub.sys=rdbecat.pub.relate
FILE rdbhelp.pub.sys=rdbhelp.pub.relate
FILE  grecat.pub.sys= grecat.pub.relate
FILE plotter;dev=50
FILE rdblist;dev=58;cctl
RUN   relate.pub.relate;lib=p;maxdata=31000

RELATE/3000 V4.40A SAT, OCT 20, 1984, 12:16 PM (C) CRI

  1)OPEN FILE NCJODAT.HISTJ;MODE=SHARED
  2)SELECT SCENARIO,CLASS,HULL,COMNUM UNIQUE BY SCENARIO,CLASS,HULL,COMNUM&
 .1&) WHERE SCENARIO="FIXIT" AND $MATCH(CLASS,"DDG")<>0
  3)PRINT CLASS,HULL
```

After giving the select JOHN asks that all the records matching his criteria be printed. He only asks for the CLASS and HULL fields, but he gets SCENARIO and COMNUN in the printout as well because those fields are part of the index he specified in the select BY clause.

He finds that there are 37 DDG construction jobs in the relation for DEMO/FIXIT. He decides this is a reasonable number and terminates his RELATE session with the "//" command.

JOHN could have been fancier and given a select command with more sophisticated clauses and conditions such that RELATE would have printed only the number "37". Going that route is a good idea when the number of records you expect is in the hundreds, but it can be tricky to get the select statement set up just right.

SCENARIO	COMNUM	CLASS	HULL
FIXIT	1	DDG-2	2
FIXIT	1	DDG-2	3
FIXIT	1	DDG-2	4
FIXIT	1	DDG-2	5
FIXIT	1	DDG-2	6
FIXIT	1	DDG-2	7
FIXIT	1	DDG-2	8
FIXIT	1	DDG-2	9
FIXIT	1	DDG-2	10
FIXIT	1	DDG-2	11
FIXIT	1	DDG-2	12
FIXIT	1	DDG-2	13
FIXIT	1	DDG-2	14
FIXIT	1	DDG-2	15
FIXIT	1	DDG-2	16
FIXIT	1	DDG-2	17
FIXIT	1	DDG-2	18
FIXIT	1	DDG-2	19
FIXIT	1	DDG-2	20
FIXIT	1	DDG-2	21
FIXIT	1	DDG-2	22
FIXIT	1	DDG-2	23
FIXIT	1	DDG-2	24
FIXIT	1	DDG-37	37
FIXIT	1	DDG-37	38
FIXIT	1	DDG-37	39
FIXIT	1	DDG-37	40
FIXIT	1	DDG-37	41
FIXIT	1	DDG-37	42
FIXIT	1	DDG-37	43
FIXIT	1	DDG-37	44
FIXIT	1	DDG-37	45
FIXIT	1	DDG-37	46
FIXIT	1	DDG-993	993
FIXIT	1	DDG-993	994
FIXIT	1	DDG-993	995
FIXIT	1	DDG-993	996

37 LINES PRINTED.
4)//

END OF PROGRAM

To be thorough, JOHN really should have gone and checked the second hypothesis (DDGs retiring too early) before leaving RELATE, but he is so confident he is right that he is going to go fix the problem right away. Since he had specified that ships be retired according to their standard service life in the force level parameters menu, he is sure that the service life specification for the DDGs must be too short. He knows that these ships are slated for SLEPs in the near future, but that actual projections of the SLEP jobs probably have not been entered in the repair jobs data base. He can simulate the effect of the SLEP jobs by just increasing the standard service lives.

To change the service lives he must get back into ALIAS. Although JOHN can look at any data in the data base using RELATE, he cannot change any that way. All changes must be made using the ALIAS Data Base Updating system (DBU). The DBU makes sure that all data base entries and changes are validated. Without validation, honest mistakes would rapidly render the ALIAS data base unusable.

So JOHN logs on with his JOHNA user name again, and gives the ALIAS command when he receives the colon prompt.

```
:HELLO JOHNA.SEA90
ENTER USER PASSWORD:

HP3000 / MPE IV C.B1.A2.  SAT, OCT 20, 1984, 12:34 PM
***************************************************************
WEEKLY BACK UP OF FILES IS NOW TAKING 8- 2400 FEET REELS
OF TAPE AND LASTING MORE THAN 2 1/2 HOURS.  IT IS IMPERATIVE
THAT USERS OF THE SYSTEM PURGE OLD FILES ON A REGULAR BASIS.
***************************************************************


Bulletin:
**********************************************************************
**********************************************************************
****************** DO NOT LEAVE YOUR TERMINAL **********************
******************          UNATTENDED         **********************
******************      SIGN    OFF   !!!!     **********************
**********************************************************************
**********************************************************************
_____________________

END OF PROGRAM

:ALIAS



                    ************************************
                    *                                  *
                    *         WELCOME TO ALIAS         *
                    *           VERSION 1.0            *
                    ************************************



                    > SYSTEM START-UP IN PROGRESS:
                     -Loading core system...
                     -Connecting to data base...
```

This time when ALIAS asks what scenario he wants to use, he can just say "DEMO". He doesn't have to create scenario DEMO again. Once a scenario is created, it stays in place until its creator or the Data Base Administrator deletes it.

```
                                          Current scenario is

Scenario choice options include:
 ?      provides help                |  ^    exits scenario choice system
 @      lists existing scenarios     |  +    moves you into scenario
 @name  lists the composition        |         creation menu
          of the 'name' scenario     |  &    re-displays this menu
 name   makes the 'name' scenario    |
        your current scenario        |
--------------------------------------------------------------------------
                       SCENARIO CHOICE MENU
Name of scenario to use, or Command character: DEMO
```

```
                *******************************
                *                             *
                *     LOADING PARAMETERS      *
                *                             *
                *******************************
```

In the Command System top menu, JOHN asks to use the DBU. If you are following along, it is important that ALIAS have made a correct guess concerning the type of terminal type you are using before you try and run the DBU. If ALIAS did not clear your display before presenting the top Command System menu, it's guess is certainly wrong and you must go and correct it in the User Environment area.

After a (probably long) delay the MASTER screen of the DBU is displayed. You should know that this DBU startup delay is a one-time cost for any given ALIAS run. When you leave the DBU, it will be put on "hold" and will be immediately ready for your later use.

Though not evident on the paper copy, the MASTER screen is subtly different from the TOP Command menu. If you are looking at it on your display, you will notice that the area after the COMMAND prompt on the display is in inverse video; it is a fill-in-the-blank area. This makes MASTER a screen, not a menu, even though functionally it is a menu of numbered choices just like TOP. As you become more familiar with ALIAS you will find that your command options are somewhat different in screens than in menus.

Notice that the fact that the DBU is a new and different context is indicated by the "ALIAS DATA BASE UPDATE SYSTEM" title at the top of the screen. As usual, when the title at the top is different, the rules will be somewhat different.

The MASTER menu presents different types of data that can be accessed and updated with the DBU. JOHN wants to change something having to do with entire DDG ship classes (their service lives), so he chooses option 1 by hitting the "1" key followed by the carriage return key.

```
Menu is TOP                  * ALIAS COMMAND SYSTEM *   Scenario is DEMO
_________________________________________________________________________
                        TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 3

Starting up Data Base Updater.  Expect a delay of several minutes.
Data maintenance module start up now in progress...
Connecting to data dictionary...
Connecting to the legal values reference library...
```

```
///////////////////////////////////////////////////////////////////////

SCREEN IS: MASTER                                    SCENARIO IS: DEMO
=========================================================================
 ? for help              ALIAS DATA BASE UPDATE SYSTEM         =NAME jumps
===============choose a screen to use by number or >NAME=================

                         COMMAND: 1

                         1. SHIP CLASSES
                         2. SHIP JOB TYPES
                         3. SHIP JOB SCHEDULES
                         4. SHIP YARDS
                         5. SHIP DEACTIVATIONS
                         6. DATA DICTIONARY UPDATING SYSTEM




=========================No data may be changed here======================
   Please place a command or option number after COMAMND and press RETURN
```

A new screen is displayed, again offering choices. JOHN wants to change a class characteristic, not schedule planning factors (the job descriptions the assigner uses in making up schedules) for a class, so again he chooses option 1.

The CLASS_CHARS data screen is presented. Its function is to display and allow changes of class descriptions. It is concerned with any data that applies to a class as a whole, which is generally physical data. In the right middle, though, there is an area labeled SHIP LIFE with a field labeled "Standard Service Time After Delivery". This is the only field JOHN will be concerned with.

If you are running a sample session as well as reading along you will note that the CLASS_CHARS screen looks much different on the display than it does on paper. This is because all the screen enhancements (the inverse video and underlining) do not show on the paper. If you are just reading be aware that there are clearly marked data-entry "windows" or "blanks" after each label (a whole matrix of rows and columns in the SHIP SIZE area) to tell the user where data should go.

JOHN wants to bring up the class descriptions for the two older DDG ship classes that showed up on his RELATE output, DDG-2 and DDG-37. There are two ways he can do this: he can work his way through all the ship classes (in alphabetical order) using the N (next) command, or he can get directly to the DDG records using the S (search) command.

The latter is much quicker for JOHN's purposes. To use it, he just puts "S" in the COMMAND field, "DDG" in the CLASS field, and hits the return key. If you are working along at the terminal, you move between fields using the TAB key. Note that the arrow keys on HP terminals will not work for these screens, but you may be able to use the terminal's function keys to achieve the same effects.

The effect of the S command is to set the DBU up to show you only records whose fields all match the values you have specified by filling them in on the screen. There is no limit to the number of fields you can fill in. The more fields you fill in, and the more precisely you fill in each one, the more precisely the records which are returned will match the target in your mind (but if you're not sure of a value, don't fill it in, because the DBU will tell you it can't find anything if such a value doesn't exist in the data base).

There's a catch to using the S command. You have to be sure that ALL the fields on the screen are blank EXCEPT for the ones you fill in as targets, including the "Entry Date" and "Entry By" fields at the bottom right (which you can't change). To make sure you start with a clean slate, give the "K" (clear) command before you start your set-up for S. The bottom frame at the right shows JOHN about to hit the return key to execute the K command.

```
SCREEN IS: CLASS_MENU                                   SCENARIO IS: DEMO
==========================================================================
 ?  for help             ALIAS DATA BASE UPDATE SYSTEM           =NAME jumps
================choose a screen to use by number or >NAME================

                         COMMAND: 1

                         1. CLASS CHARACTERISTICS
                         2. SCHEDULE PLANNING FACTORS




======================No data may be changed here=========================
    Please place a command or option number after COMAMND and press RETURN
```

//

```
SCREEN IS: CLASS_CHARS          LATEST  DATA          SCENARIO IS: DEMO
==========================================================================
 ?  for help             ALIAS DATA BASE UPDATE SYSTEM           =NAME jumps
=======================Ship Class Characteristics=========================

                         COMMAND: K

       Class Name:                                  Owner:   USN

   Name for Reports:

                    SHIP SIZE                               SHIP LIFE
                    ---------                               ---------
            Length Beam   Height  Draft  Displacement    Standard Service Time
Waterline:                                               After Delivery:
Overall:                                                 in Time Units:  MONTHS
At Launch:                                               Data Source
Light:                                                   Data Date
Full Load:                                               Entry Date
                                                         Entry By
==============Your priveleges in this screen are: inspect==================
               Place a command after COMMAND and press RETURN
```

The top frame at the right shows the display after K has been executed. Notice that the values which were showing in the "Owner" and "Time Units" fields on the previous frame have been erased.

JOHN is then free to fill "DDG" into the class field and to give the S command.

```
SCREEN IS: CLASS_CHARS            LATEST  DATA        SCENARIO IS: DEMO
==========================================================================
 ?  for help               ALIAS DATA BASE UPDATE SYSTEM          =NAME jumps
=======================Ship Class Characteristics=========================

                          COMMAND:

       Class Name:                            Owner:

  Name for Reports:

                      SHIP SIZE                                SHIP LIFE
                      ---------                                ---------

          Length Beam   Height  Draft  Displacement     Standard Service Time
Waterline:                                              After Delivery:
Overall:                                                in Time Units:
At Launch:                                              Data Source
Light:                                                  Data Date
Full Load:                                              Entry Date
                                                        Entry By
==============Your privileges in this screen are: inspect=================
              Place a command after COMMAND and press RETURN
```

//

```
SCREEN IS: CLASS_CHARS            LATEST  DATA        SCENARIO IS: DEMO
==========================================================================
 ?  for help               ALIAS DATA BASE UPDATE SYSTEM          =NAME jumps
=======================Ship Class Characteristics=========================

                          COMMAND: S

       Class Name:     DDG                    Owner:

  Name for Reports:

                      SHIP SIZE                                SHIP LIFE
                      ---------                                ---------

          Length Beam   Height  Draft  Displacement     Standard Service Time
Waterline:                                              After Delivery:
Overall:                                                in Time Units:
At Launch:                                              Data Source
Light:                                                  Data Date
Full Load:                                              Entry Date
                                                        Entry By
==============Your privileges in this screen are: inspect=================
              Place a command after COMMAND and press RETURN
```

S returns the first matching record (if it can find one). The first match is on DDG-2 in this case, which is one of the two classes JOHN is interested in. The job description is rather incomplete, (having no ship size information at all!), but does have a standard service life specification of 30 calendar years. This is about what JOHN expected to see, since most ships without SLEPs last about this amount of time.

In the bottom frame JOHN has changed the standard life to 45 years to simulate the SLEPs he expects, and has changed the Data Source to "IMAGINED" as a reminder that this is not "real" data (notice that on the paper copies of the screens you can distinguish between what JOHN types and what ALIAS types by the fact that JOHN's entries are in `bold face`).

He has just executed the M (modify) command, which will replace the old record with the new in the data base, but is chagrined to see an error message appear at the bottom of the screen: "Scenario security forbids your making data changes here." He is chagrined because all along the screen has been telling him in the privilege-status line two lines up that "Your priveleges in this screen are: inspect", i.e. that he can look at data but can't make changes.

Why has this happened? There are two reasons why his data changes privileges might be limited in a screen. One is that the Data Base Administrator may have set up security such that JOHN is <u>never</u> able to change data in the CLASS_CHARS screen. The other is that JOHN may have set up his scenario so he is using the class-description data (stored in group .MISCJ) indirectly. Data being used indirectly may not be changed, since it really belongs to someone else's scenario.

The error message is the tip-off that this is the case. If you look back to JOHN's creation of scenario DEMO in the first few pages of this Section, you will see that he indeed answered "No" when asked if he expected to make changes to data in the .MISCJ group.

```
SCREEN IS: CLASS_CHARS              LATEST  DATA         SCENARIO IS: DEMO
============================================================================
 ?  for help                 ALIAS DATA BASE UPDATE SYSTEM           =NAME jumps
=======================Ship Class Characteristics===========================

                              COMMAND:

        Class Name:        DDG-2                   Owner:   USN

     Name for Reports:

                          SHIP SIZE                              SHIP LIFE

             Length Beam    Height  Draft  Displacement     Standard Service Time
Waterline:    0      0                                      After Delivery: 30
Overall:      0      0                                      in Time Units:  CALYR
At Launch:                   0       0       0              Data Source CSDS
Light:                       0       0       0              Data Date    3/28/1984
Full Load:                   0       0       0              Entry Date  3/28/1984
                                                            Entry By     MARK
===============Your privileges in this screen are: inspect==================
              Place a command after COMMAND and press RETURN
```

//

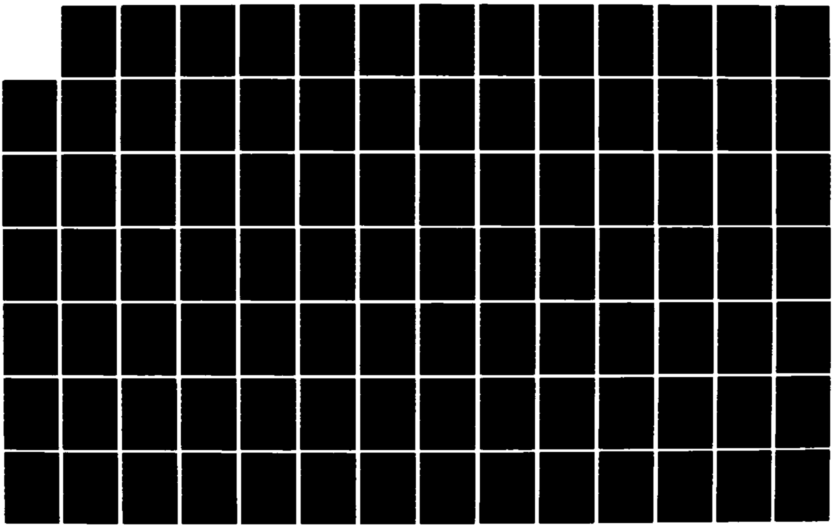
```
SCREEN IS: CLASS_CHARS              LATEST  DATA         SCENARIO IS: DEMO
============================================================================
 ?  for help                 ALIAS DATA BASE UPDATE SYSTEM           =NAME jumps
=======================Ship Class Characteristics===========================

                              COMMAND: M

        Class Name:        DDG-2                   Owner:   USN

     Name for Reports:

                          SHIP SIZE                              SHIP LIFE

             Length Beam    Height  Draft  Displacement     Standard Service Time
Waterline:    0      0                                      After Delivery: 45
Overall:      0      0                                      in Time Units:  CALYR
At Launch:                   0       0       0              Data Source IMAGINED
Light:                       0       0       0              Data Date    3/28/1984
Full Load:                   0       0       0              Entry Date  3/28/1984
                                                            Entry By     MARK
===============Your privileges in this screen are: inspect==================
              Place a command after COMMAND and press RETURN
Scenario security forbids your making data changes here.
```

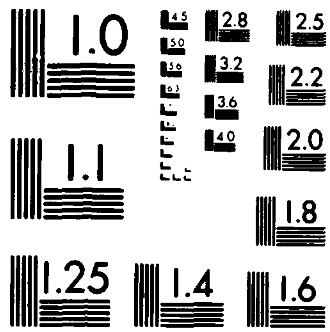
What's to be done? Does JOHN have to start over? Certainly not, this is no more than a minor annoyance. He need only use the scenario system's capability to change scenario compositions to convert .MISCJ data for scenario DEMO from indirect to direct-access status. To do this, JOHN asks to Quit (Q) from the DBU.

He is returned to the Command System TOP menu (from whence he called up the DBU), and asks for option 6, the scenario system.

AD-A150 424   ACQUISITION AND LOGISTICS INFORMATION AND ANALYSIS SYSTEM (ALIAS) USER'S GUIDE(U) DECISION-SCIENCE APPLICATIONS INC ARLINGTON VA M S CAREY ET AL.   2/4

UNCLASSIFIED   31 OCT 84 DSA-618 N00014-82-C-0813   F/G 15/5   NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
SCREEN IS: CLASS_CHARS            LATEST  DATA        SCENARIO IS: DEMO
==========================================================================
 ?  for help              ALIAS DATA BASE UPDATE SYSTEM            =NAME jumps
=======================Ship Class Characteristics=========================

                          COMMAND: Q

      Class Name:        DDG-2                   Owner:    USN

    Name for Reports:

                        SHIP SIZE                              SHIP LIFE

               Length Beam    Height  Draft  Displacement    Standard Service Time
Waterline:     0      0                                      After Delivery: 45
Overall:       0      0                                      in Time Units:   CALYR
At Launch:                    0       0      0               Data Source IMAGINED
Light:                        0       0      0               Data Date    3/28/1984
Full Load:                    0       0      0               Entry Date   3/28/1984
                                                             Entry By     MARK
===============Your privileges in this screen are: inspect================
             Place a command after COMMAND and press RETURN

//////////////////////////////////////////////////////////////////////

Menu is TOP                 * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
                           TOP LEVEL ALIAS COMMAND MENU
   1.  CUSTOMIZE USER ENVIRONMENT
   2.  CALL NON-ALIAS PROCESSORS
   3.  DATA BASE UPDATING SYSTEM
   4.  MANUAL ASSIGNMENT EDITOR
   5.  FORCE LEVEL REPORT GENERATOR
   6.  SCENARIO CHOICE/MAKEUP SYSTEM

 COMMAND: 6
```

In the Command Systems SCEN choice menu, he wants to "Modify the makeup of an existing scenario." The scenario system's modification menu (stage 1) menu comes up in response to his request for option 6. Notice that we are in yet another context here: there is no real menu title, only a header line noting that JOHN is in the scenario system and the activity he is performing.

In this "stage 1" menu JOHN must name the scenario he wants to change the composition of (which can be different from the one he is running). He wants to find out the current composition of DEMO before he changes it, though, so he gives the "DEMO" command (listed as "name" at the top of the menu as one of the available service commands).

SCENARIO SYSTEM MENU

1. CHOOSE A DIFFERENT SCENARIO TO WORK WITH
2. CREATE A NEW SCENARIO
3. DELETE CURRENTLY EXISTING SCENARIOS
4. LIST CURRENTLY EXISTING SCENARIOS
5. SEND LIST OF EXISTING SCENARIOS TO LINE PRINTER
6. MODIFY THE MAKEUP OF AN EXISTING SCENARIO

COMMAND: 6

//

Scenario modification options include:

?	provides help	^	exits scenario choice system
@	lists existing scenarios	&	re-displays this menu
name	specifies the name of the scenario to modify	@name	lists the composition of the 'name' scenario

CHANGE SCENARIO MAKEUP—STAGE 1

NAME of scenario to modify, or COMMAND: @DEMO

He is asked if he wants to know the composition by individual data base relation, or by family/group. He chooses the latter and ALIAS prints out a list of groups similar to the one he was asked about during scenario creation, along with the name of the scenario the data is currently coming from (these names are the values the SCENARIO field takes on in the DB relations in the respective groups). In this case, a value of FIXIT indicates that the data is indirect-access, and DEMO that it is direct access. The .MISCJ data indeed is of the indirect access variety for scenario DEMO.

The scenario system prompts again for a command or the name of the scenario to modify. JOHN, now ready, responds with the name.

In the next frame (stage 2) JOHN's fundamental choice is the level at which the change will be made. To change the status of an individual relation's data the "CR" command should be given; to change the status of a whole group/family the "CS" command is required. It is generally wise to make changes on an entire-group basis, so JOHN chooses CS.

```
Do you want the composition by relation (no=by family)? n



                 SCENARIOS CONTRIBUTING TO SCENARIO DEMO

RELATION FAMILY        USES DATA FROM
_______________        ______________
CURRJ                  FIXIT
DESCJ                  DEMO
HISTJ                  FIXIT
MISCJ                  FIXIT
PARAMS                 DEMO
PROJJ                  DEMO
YARDS                  FIXIT
NAME of scenario to modify, or COMMAND: DEMO


//////////////////////////////////////////////////////////////////////


               *** MODIFYING MAKEUP OF SCENARIO DEMO ***
Scenario modification options include:
 ?      provides help                |  ^     exits scenario choice system
 CS     change composition by family |  LS    list composition by family
 CR     change composition by file   |  LF    list composition by file
 &      re-displays this menu        |
______________________________________________________________________________
                    CHANGE SCENARIO MAKEUP--STAGE 2
 COMMAND: cs
```

In stage three JOHN specifies the changes he wants made. This is done using an equation-like syntax, as the prompt indicates. In this particular case what JOHN does seems a little redundant: he asks that the new_source_scenario for the .MISCJ group's data be FIXIT, but the source scenario was already FIXIT! His goal is to keep using the FIXIT data, but directly rather than indirectly. Thus he really just wants to answer "Yes" to the "Do you want a data copy made" question, which is identical in meaning to the "Will you want to change the data in that group?" question asked during scenario creation.

A pause follows while the data copy is done; then the "Change made." message appears. Since this is the only thing JOHN wanted to do, he "pops" using the "^" command, pops again back into the Command System SCEN menu, and jumps to the TOP menu (with /).

```
Scenario modification options include:
 ?      provides help            | ^      exits scenario choice system
 @      lists current scenarios  | @name lists composition of NAME
 &      re-displays this menu    | FAMILY=NEW SOURCE specifies change
------------------------------------------------------------------------
                    CHANGE SCENARIO MAKEUP--STAGE 3
DB_FAMILY=NEW_SOURCE_SCENARIO, or COMMAND: MISCJ=FIXIT

Do you want a copy of that data made (no=only read it)? Y

Change made.
DB_FAMILY=NEW_SOURCE_SCENARIO, or COMMAND: ^

 COMMAND: ^
```

//

```
Menu is SCEN             * ALIAS COMMAND SYSTEM *   Scenario is DEMO
------------------------------------------------------------------------
                         SCENARIO SYSTEM MENU
   1.  CHOOSE A DIFFERENT SCENARIO TO WORK WITH
   2.  CREATE A NEW SCENARIO
   3.  DELETE CURRENTLY EXISTING SCENARIOS
   4.  LIST CURRENTLY EXISTING SCENARIOS
   5.  SEND LIST OF EXISTING SCENARIOS TO LINE PRINTER
   6.  MODIFY THE MAKEUP OF AN EXISTING SCENARIO

  COMMAND: /
```

JOHN asks for the DBU again. Users running at a terminal will notice that its response is almost immediate, in comparison to the long startup delay the first time it was requested.

Screen MASTER comes up rather than the CLASS_CHARS screen JOHN was running. He can't jump right back to that screen because he might have changed the scenario he was using while out of the DBU, and DBU screens set up scenario security during their set-up phase.

Wanting to return to the CLASS_CHARS screen, JOHN remembers its name since he was just using it. When you know the name of the screen you want to use in the DBU, you can jump to it directly by giving the "=screen_name" command instead of working your way down the chain of menus. JOHN has set up and executed the appropriate command in the MASTER screen in the bottom frame opposite.

```
Menu is TOP                * ALIAS COMMAND SYSTEM *   Scenario is DEMO
--------------------------------------------------------------------------
                        TOP LEVEL ALIAS COMMAND MENU
   1.  CUSTOMIZE USER ENVIRONMENT
   2.  CALL NON-ALIAS PROCESSORS
   3.  DATA BASE UPDATING SYSTEM
   4.  MANUAL ASSIGNMENT EDITOR
   5.  FORCE LEVEL REPORT GENERATOR
   6.  SCENARIO CHOICE/MAKEUP SYSTEM

 COMMAND: 3

Back in the DBU now.
```

//

```
SCREEN IS: MASTER                                  SCENARIO IS: DEMO
==========================================================================
 ? for help           ALIAS DATA BASE UPDATE SYSTEM           =NAME jumps
===================choose a screen to use by number or >NAME==============

                        COMMAND: =CLASS_CHARS

                        1. SHIP CLASSES
                        2. SHIP JOB TYPES
                        3. SHIP JOB SCHEDULES
                        4. SHIP YARDS
                        5. SHIP DEACTIVATIONS
                        6. DATA DICTIONARY UPDATING SYSTEM

=========================No data may be changed here======================
    Please place a command or option number after COMAMND and press RETURN

Getting screen...
```

JOHN has to repeat the K-S (clear-search sequence) to get back the DDG-2 record he was looking at. Notice that the privilege line at the bottom of the screen now says that JOHN has the full list of data change privileges, now that he is working with his "own" copy of class characteristics data.

```
SCREEN IS: CLASS_CHARS          LATEST  DATA        SCENARIO IS: DEMO
=========================================================================
 ?  for help            ALIAS DATA BASE UPDATE SYSTEM           =NAME jumps
========================Ship Class Characteristics=======================

                        COMMAND: K

          Class Name:                          Owner:   USN

    Name for Reports:

                    SHIP SIZE                               SHIP LIFE

            Length Beam   Height  Draft  Displacement   Standard Service Time
Waterline:                                              After Delivery:
Overall:                                                in Time Units:  MONTHS
At Launch:                                              Data Source
Light:                                                  Data Date
Full Load:                                              Entry Date  4/19/1984
                                                        Entry By    MARK
 ===Your privileges in this screen are: inspect, add, modify, update, delete===
              Place a command after COMMAND and press RETURN
```

//

```
SCREEN IS: CLASS_CHARS          LATEST  DATA        SCENARIO IS: DEMO
=========================================================================
 ?  for help            ALIAS DATA BASE UPDATE SYSTEM           =NAME jumps
========================Ship Class Characteristics=======================

                        COMMAND: S

          Class Name:     DDG                  Owner:

    Name for Reports:

                    SHIP SIZE                               SHIP LIFE

            Length Beam   Height  Draft  Displacement   Standard Service Time
Waterline:                                              After Delivery:
Overall:                                                in Time Units:
At Launch:                                              Data Source
Light:                                                  Data Date
Full Load:                                              Entry Date
                                                        Entry By
 ===Your privileges in this screen are: inspect, add, modify, update, delete===
              Place a command after COMMAND and press RETURN
```

He sets up his modification again (45 year life rather than 30), and this time decides to change the data date as well. Changing the data date makes it necessary for him to use the U (update) command rather than the M (modify) command to save his change into the data base. Instead of just writing over the old record, Update adds a record with JOHN's changes to the appropriate file(s), preserving the old record in case he wants to get back to it. Updating is the most appropriate act in this case since JOHN's change is really temporary in nature.

In the screen showing in the bottom frame, JOHN has already brought up the description for DDG-37 and done an Update (notice the service life, entry date, and entry by field values. He is preparing to "rewind" back to the start of the file using the "B" (back to the beginning) command so that he can look and verify that his changes were made.

```
SCREEN IS: CLASS_CHARS              LATEST  DATA          SCENARIO IS: DEMO
===========================================================================
 ?  for help                ALIAS DATA BASE UPDATE SYSTEM              =NAME jumps
========================Ship Class Characteristics==========================

                          COMMAND: U

            Class Name:       DDG-2                   Owner:    USN

        Name for Reports:

                        SHIP SIZE                                  SHIP LIFE
                        ---------                                  ---------
                 Length Beam   Height  Draft  Displacement   Standard Service Time
Waterline:         0     0                                   After Delivery: 45
Overall:           0     0                                   in Time Units:   CALYR
At Launch:                       0       0       0           Data Source IMAGINED
Light:                           0       0       0           Data Date    10/15/1984
Full Load:                       0       0       0           Entry Date    3/28/1984
                                                             Entry By     MARK
 ===Your privileges in this screen are: inspect, add, modify, update, delete===
                 Place a command after COMMAND and press RETURN

Want comments associated with old record brought forward? (Y):N

/////////////////////////////////////////////////////////////////

SCREEN IS: CLASS_CHARS              LATEST  DATA          SCENARIO IS: DEMO
===========================================================================
 ?  for help                ALIAS DATA BASE UPDATE SYSTEM              =NAME jumps
========================Ship Class Characteristics==========================

                          COMMAND: B

            Class Name:       DDG-37                  Owner:    USN

        Name for Reports:

                        SHIP SIZE                                  SHIP LIFE
                        ---------                                  ---------
                 Length Beam   Height  Draft  Displacement   Standard Service Time
Waterline:         0     0                                   After Delivery: 45
Overall:           0     0                                   in Time Units:   CALYR
At Launch:                       0       0       0           Data Source CSDS
Light:                           0       0       0           Data Date  10/16/1984
Full Load:                       0       0       0           Entry Date 10/16/1984
                                                             Entry By     JOHNA
 ===Your privileges in this screen are: inspect, add, modify, update, delete===
                 Place a command after COMMAND and press RETURN
```

A few steps have not been shown to save space here. JOHN has repeated the clear-search sequence for DDGs, which has brought up the description for DDG-2. Note that it is JOHN's updated description---the change really was made. He is giving the N (next) command to look to make sure the DDG-37 class was changed as well. The bottom frame verifies that it was.

He does another N to see what comes next.

SCREEN IS: CLASS_CHARS LATEST DATA SCENARIO IS: DEMO

? for help ALIAS DATA BASE UPDATE SYSTEM =NAME jumps

Ship Class Characteristics

COMMAND: N

Class Name: DDG-2 Owner: USN

Name for Reports:

SHIP SIZE

	Length	Beam	Height	Draft	Displacement
Waterline:	0	0			
Overall:	0	0			
At Launch:			0	0	0
Light:			0	0	0
Full Load:			0	0	0

SHIP LIFE

Standard Service Time
After Delivery: 45
in Time Units: CALYR
Data Source IMAGINED
Data Date 10/15/1984
Entry Date 10/26/1984
Entry By JOHNA

===Your privileges in this screen are: inspect, add, modify, update, delete===
Place a command after COMMAND and press RETURN

//

SCREEN IS: CLASS_CHARS LATEST DATA SCENARIO IS: DEMO

? for help ALIAS DATA BASE UPDATE SYSTEM =NAME jumps

Ship Class Characteristics

COMMAND: N

Class Name: DDG-37 Owner: USN

Name for Reports:

SHIP SIZE

	Length	Beam	Height	Draft	Displacement
Waterline:	0	0			
Overall:	0	0			
At Launch:			0	0	0
Light:			0	0	0
Full Load:			0	0	0

SHIP LIFE

Standard Service Time
After Delivery: 45
in Time Units: CALYR
Data Source CSDS
Data Date 10/16/1984
Entry Date 10/26/1984
Entry By JOHNA

===Your privileges in this screen are: inspect, add, modify, update, delete===
Place a command after COMMAND and press RETURN

Finding DDG-51 next, and knowing he has changed the description for the relevant classes from ncjodat.histj, he asks to leave the DBU again. Back in the TOP Command System menu again, he is ready for another battle group report generator run to find out the effect of his change.

```
SCREEN IS: CLASS_CHARS              LATEST  DATA          SCENARIO IS: DEMO
================================================================================
 ?  for help                  ALIAS DATA BASE UPDATE SYSTEM            =NAME jumps
=========================Ship Class Characteristics============================

                              COMMAND: Q

          Class Name:         DDG-51                   Owner:   USN

     Name for Reports:

                          SHIP SIZE                                 SHIP LIFE
                          ---------                                 ---------
              Length Beam    Height  Draft  Displacement     Standard Service Time
Waterline:      0      0                                     After Delivery: 30
Overall:        0      0                                     in Time Units:  CALYR
At Launch:                     0       0       0             Data Source CSDS
Light:                         0       0       0             Data Date    3/28/1984
Full Load:                     0       0       0             Entry Date   3/28/1984
                                                             Entry By     MARK
 ===Your privileges in this screen are: inspect, add, modify, update, delete===
                 Place a command after COMMAND and press RETURN

////////////////////////////////////////////////////////////////////////

Menu is TOP                       * ALIAS COMMAND SYSTEM *   Scenario is DEMO
--------------------------------------------------------------------------------
                            TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 5
```

JOHN executes the battle group report generator again using the same format control file (BGPOM86.FMTFIL). He has to give a different name for the file that the report will be saved in than last time, though, since the first report is already in the BGDEMO file. He chooses BGREPT as the repository for the second try's output.

Notice that JOHN did not have to go modify the parameters and list of out-of-force repair jobs again, as he did the first time. The changes he made will remain in effect (for the DEMO scenario) until he decides he wants to change them again, even if he logs off. This is one of the most convenient features of ALIAS---once you get things set up, you don't have to worry about them until you want the set-up changed.

The battle group report generator provides its usual progress messages as it does its work. JOHN takes a break.

An important thing to reflect on at this point is the fact that JOHN made a mistake in setting up his scenario (he didn't realize that he would want to change ship life data, and so during the DEMO set-up made that data indirect-access), and yet was able to recover from the mistake without much trouble. This is generally true in ALIAS---few mistakes are so serious that they cannot be undone. The best way to use ALIAS is to concentrate on the analytical content of the decisions you make without worrying too much about getting everything right the first time. It _is_ important to review all of the relevant system control values, like the contents of parameter menus, since they will affect your results, but there is no harm done by trying several different settings.

Serious problems _can_ result from a loss of data base integrity, but as long as everyone uses the DBU for data base updating the chance of integrity loss is minimized.

FORCE LEVEL GENERATOR

1. FORCE LEVEL REPORT INITIALIZATION PARAMETERS
2. EXECUTE FORCE REPORT GENERATOR
3. PREPARE BATTLE GROUP REPORT

```
COMMAND: 3

Starting up Battle Group Report Generator. Please stand by.
NAME OF OUTPUT CONTROL FILE (name.group, or ^) BGPOM86.FMTFIL

The file FLREPT already exists in your group.
The force level modules usually stores its report in that
file when you request that the report be saved on disk.
Since that file is in use, what file would you like to store
the report in? (name <= 8 letters, or / to not save report): BGREPT




////////////////////////////////////////////////////////////////////


Opening required data base files...
Looking for ships in data base...
Looking for class BB-61
Looking for class CG-16
Looking for class CG-26
Looking for class CG-47
Looking for class CGN-25
Looking for class CGN-35
Looking for class CGN-36
Looking for class CGN-38
Looking for class CGN-9
Looking for class CV-41
Looking for class CV-59
Looking for class CV-63
Looking for class CV-67
Looking for class CVN-65
Looking for class CVN-68
Looking for class DD-963
Looking for class DDG-2
Looking for class DDG-37
Looking for class DDG-51
Looking for class DDG-993
Looking for class FF-1040
Looking for class FF-1052
Looking for class FFG-1
Looking for class FFG-7
```

The resulting battle group report still shows mixed results. The number of carrier battle groups available in the 1990's has improved substantially with the change in DDG service life estimates. There is seemingly even a small surplus of DDGs. Now the constraint on battle group formation would seem to be cruiser and DD availability. There are still "excess" carriers, enough to reach the 17 CVBG goal. And there is still the puzzling "excess" of carriers in the near term.

JOHN decides that the first order of business is to build more cruisers and DD's. He is left in the Force Level choice menu when report production finishes, and asks for a pop to the top menu from there.

DEPLOYABLE BATTLE GROUP PROJECTION FOR POM-86
BASED ON SURFACE COMBATANT REQUIREMENTS ONLY
(ALL DATA NOTIONAL)

CALENDAR YEAR	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
BATTLEGROUP														
CARRIER BG	9	9	9	9	9	10	11	13	14	15	15	13	13	13
SURFACE AG														
MARINE AF				1	1									
SUPPLY ESCORT	1	1	1			1	1		3	1	1	5	5	5
CONVOY	10	10	10	5	5	10	8	5						
BALANCE														
CARRIER	4	4	4	5	5	5	3	2	2	1	1	4	4	4
BB			1	1	1	1	1	1	1					
CRUISER	7	7	7	5	5	6	3	1						
DDG										1	1	5	5	5
DD	2	2	2											
FFG														
FF	33	33	34	46	46	30	34	40	49	45	43	38	33	28

//

Menu is FLRPTG * ALIAS COMMAND SYSTEM * Scenario is DEMO

FORCE LEVEL GENERATOR

1. FORCE LEVEL REPORT INITIALIZATION PARAMETERS
2. EXECUTE FORCE REPORT GENERATOR
3. PREPARE BATTLE GROUP REPORT

COMMAND: /

To change his POM JOHN needs to run the assigner again. He moves into the Assigner choice menu, but instead of running it immediately he asks for the assigner parameter menu again.

```
Menu is TOP                * ALIAS COMMAND SYSTEM *   Scenario is DEMO
----------------------------------------------------------------------
                    TOP LEVEL ALIAS COMMAND MENU
   1.  CUSTOMIZE USER ENVIRONMENT
   2.  CALL NON-ALIAS PROCESSORS
   3.  DATA BASE UPDATING SYSTEM
   4.  MANUAL ASSIGNMENT EDITOR
   5.  FORCE LEVEL REPORT GENERATOR
   6.  SCENARIO CHOICE/MAKEUP SYSTEM

 COMMAND: 4
```

//

```
Menu is ASSIGN             * ALIAS COMMAND SYSTEM *   Scenario is DEMO
----------------------------------------------------------------------
                    MANUAL ASSIGNER SPECIFICATIONS
   1.  ASSIGNER INITIALIZATION PARAMETERS
   2.  EXECUTE THE ASSIGNER

 COMMAND: 1
```

His motivation is to save some time by reducing the amount of data both he and the assigner will have to process. He knows that during this assigner run he will change assignments only at a subset of the yards involved in producing the POM---only at those which he will have building CG's and DDG's. He knows already that there will be only four such yards: BIW, INGALLS, TODD LA, and TODD SEA. By "turning off" all the other yards he can make the assigner load and present assignments only for those four. This will take less time, and he will not have to "thumb" through several pages of assignments on the screen.

To put this into effect JOHN asks to see the list of CANDIDATE SHIP YARDS in the parameter menu. The first page of the list is displayed in the bottom frame opposite. Notice that the yards are all "turned on"---that is, they all have an asterisk next to their name. JOHN's first act is to turn them all off with the "N" (none) command, and then to turn BIW back on by giving its number (21). Then he asks for the next page of the list.

```
Menu is ASNPRM              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
                MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                 = FISCYR        (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE             = 10/ 1/1985    (MM/DD/YYYY)
 3.  ENDING DATE               =  9/31/1994    (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS      = LIST          (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES    = LIST          (ALL/LIST)
 6.  CANDIDATE JOB TYPES       = LIST          (ALL/LIST)
 7.  DISPLAY BASIS             = AWD           (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS              = START         (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE               = PROGRAM       (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION         = PROJ          (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER      = ALPHABETIC    (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER       = ALPHABETIC    (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH              = OFF           (ON,OFF)

  COMMAND: 4=LIST
```

//

```
Menu is CHSYDS              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
          CHOOSE THE SET OF VALID YARDS TO WHICH SHIPS MAY BE ASSIGNED
---------------------------------------------------------------------------
   1. * A&E IND                        13. * BELLING
   2. * AAA SF                         14. * BETH BA
   3. * AAA SO                         15. * BETH KEY
   4. * ADDSCO                         16. * BETH NJ
   5. * ALLIED                         17. * BETH SF
   6. * AMSHIP T                       18. * BETH SP
   7. * ARCWEL                         19. * BETHBEAU
   8. * ATKINSON                       20. * BETHBOST
   9. * ATLAN DD                       21. * BIW
  10. * AVONDALE                       22. * BIW PORT
  11. * BAYSHIP                        23. * BOEINGFL
  12. * BELLHALT                       24. * BOEINGSE
---------------------------------------------------------------------------

  COMMAND: N

  COMMAND: 21

  COMMAND: +
```

Note how the N command turned off everything in the following pages. JOHN turns INGALLS on as he works his way down the list.

Menu is CHSYDS * ALIAS COMMAND SYSTEM * Scenario is DEMO

CHOOSE THE SET OF VALID YARDS TO WHICH SHIPS MAY BE ASSIGNED

25.	BOLAND	37.	GEN SHIP
26.	BRASWELL	38.	HIGGINS
27.	BROOKE	39.	HOBOKEN
28.	CHASNNSY	40.	HORNE
29.	COASTAL	41.	HP NSY
30.	DEFOE	42.	INGALLS
31.	DETYENS	43.	JACK ENG
32.	DILLINHM	44.	JACKSHIP
33.	EB GROT	45.	JONATHAN
34.	EB QP	46.	LAKEUNIN
35.	GDQ	47.	LARSONBT
36.	GDQ CHAR	48.	LB NSY

COMMAND: 42

COMMAND: +

//

Menu is CHSYDS * ALIAS COMMAND SYSTEM * Scenario is DEMO

CHOOSE THE SET OF VALID YARDS TO WHICH SHIPS MAY BE ASSIGNED

49.	LOCKHEED	61.	NEWPORTS
50.	MARE NSY	62.	NNEWS
51.	MARINET	63.	NORF NSY
52.	MARINP&E	64.	NORSHIP
53.	MARTINAC	65.	NW MARIN
54.	MARTINLI	66.	NY NSY
55.	MARYSHIP	67.	NY SHIP
56.	MERRITT	68.	PACIFIC
57.	METAL TR	69.	PENNSHIP
58.	METRO	70.	PERTH AM
59.	MUNRO DD	71.	PETERSON
60.	NASSCO	72.	PH NSY

COMMAND: +

On the last page of the list he turns on the two Todd yards, pops back up to the parameter menu, and thence back to the assigner choice menu.

```
Menu is CHSYDS              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
----------------------------------------------------------------------------
          CHOOSE THE SET OF VALID YARDS TO WHICH SHIPS MAY BE ASSIGNED
----------------------------------------------------------------------------
   73.    PHIL NSY                        83.    SWMAR SP
   74.    PTSMHNSY                        84.    SWYGERT
   75.    PUGETNSY                        85.    TACOMA
   76.    SD IR&ST                        86.    TODD ALA
   77.    SER ENG                         87.    TODD LA
   78.    SF NSY                          88.    TODD SEA
   79.    SF WELD                         89.    TODD SF
   80.    SOUTHPRT                        90.    UNKBLDR
   81.    SWMAR SD                        91.    WILIR&ST
   82.    SWMAR SF
----------------------------------------------------------------------------

  COMMAND: 87

  COMMAND: 88

  COMMAND: ^

Saving list on/off settings...


///////////////////////////////////////////////////////////////////////


Menu is ASNPRM              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
----------------------------------------------------------------------------
               MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                = FISCYR       (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE            = 10/ 1/1985   (MM/DD/YYYY)
 3.  ENDING DATE              =  9/31/1994   (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS     = LIST         (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES   = LIST         (ALL/LIST)
 6.  CANDIDATE JOB TYPES      = LIST         (ALL/LIST)
 7.  DISPLAY BASIS            = AWD          (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS             = START        (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE              = PROGRAM      (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION        = PROJ         (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER     = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER      = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH             = OFF          (ON,OFF)

  COMMAND: ^

Saving parameter settings...
```

He runs the assigner now. Notice that only the four yards he turned on are loaded.

MANUAL ASSIGNER SPECIFICATIONS

1. ASSIGNER INITIALIZATION PARAMETERS
2. EXECUTE THE ASSIGNER

COMMAND: 2

```
Starting up Manual Ship Assigner.  Expect a one to five minute delay.
Loading assignments for yard BIW
Loading assignments for yard INGALLS
Loading assignments for yard TODD LA
Loading assignments for yard TODD SEA
```

Notice also that all the assignments for these yards fit on a single page, making them somewhat easier to work with. JOHN adds two more CG-47's to the BIW yard with two Modify commands (he could have done it with one, but he hit the return key by accident in the middle of doing the first one). He asks for a refresh of the display (& command).

```
Scenario: DEMO                    *SHIP ASSIGNMENTS*   Page  1A   Time in: FISCYR
Yard      Period:| 1  2  3  4  5  6  7  8  9                                  |  9
   Shipclass T|86 87 88 89 90 91 92 93 94                                     |TOT
BIW          #01|--+--+--+--+--+--+--+--+--|----------------------------------|---
  1 CG-47       | 1  1  1  2                                                  |  5
  2 DDG-51      |    Y1 F2  1  2                                              |  6
INGALLS      #02|--+--+--+--+--+--+--+--+--|----------------------------------|---
  1 BB-61      r|    1                                                        |  1
  2 CG-47       | 2  2  2  2                                                  |  8
  3 DDG-51      |    F1  2  2  2                                              |  7
  4 LHD         |       L2  1  1                                              |  4
TODD LA      #03|--+--+--+--+--+--+--+--+--|----------------------------------|---
  1 DDG-51      |    Y1  2  2  2                                              |  7
TODD SEA     #04|--+--+--+--+--+--+--+--+--|----------------------------------|---
  1 AFDM        |        1     1                                              |  2
  2 DDG-51      |       Y1 F2  1                                              |  4
                |--+--+--+--+--+--+--+--+--|----------------------------------|---
  4    9  TOTALS| 3  7 13 12  9                                               | 44
(?=help) > M 1.1

Modify assignment to yard BIW
(Blanks denote no change; use zero for assignment deletion)
          Period:| 1  2  3  4  5  6  7  8  9
   Shipclass T|86 87 88 89 90 91 92 93 94
  1 CG-47       | 1  1  1  2
                  .. .. .. .. .. .. .. .. ..
                      2

(?=help) > M 1.1

Modify assignment to yard BIW
(Blanks denote no change; use zero for assignment deletion)
          Period:| 1  2  3  4  5  6  7  8  9
   Shipclass T|86 87 88 89 90 91 92 93 94
  1 CG-47       | 1  2  1  2
                  .. .. .. .. .. .. .. .. ..
                                 1

(?=help) > &
```

Now he changes the INGALLS assignments so that the yard is to build two more CG-47's and three more DDG-51's. He puts in the extra DDG-51's because the "surplus" of them shown on the last battle group report is an illusion---as soon as there are more cruisers and DD's available, additional DDG's will also be needed to make up battle groups.

JOHN asks to add assignments in a new ship class to INGALLS with the "A 2" command. These are DD-963 assignments, but he hits the return key by mistake again before he has the numbers to be awarded in each year typed in. The assigner issues a warning saying he didn't make any awards for the new class. He asks for a refresh to see what the result of the error is.

```
Scenario: DEMO              *SHIP ASSIGNMENTS*   Page  1A   Time in: FISCYR
Yard      Period:| 1  2  3  4  5  6  7  8  9                   | 9
   Shipclass T|86 87 88 89 90 91 92 93 94                      |TOT
BIW            #01|--+--+--+--+--+--+--+--+--|-----------------|---
  1 CG-47         | 1  2  1  2  1                              |  7
  2 DDG-51        |   Y1 F2  1  2                              |  6
INGALLS        #02|--+--+--+--+--+--+--+--+--|-----------------|---
  1 BB-61        r|    1                                       |  1
  2 CG-47         | 2  2  2  2                                 |  8
  3 DDG-51        |   F1  2  2  2                              |  7
  4 LHD           |      L2  1  1                              |  4
TODD LA        #03|--+--+--+--+--+--+--+--+--|-----------------|---
  1 DDG-51        |   Y1  2  2  2                              |  7
TODD SEA       #04|--+--+--+--+--+--+--+--+--|-----------------|---
  1 AFDM          |       1     1                              |  2
  2 DDG-51        |      Y1 F2  1                              |  4
                  |--+--+--+--+--+--+--+--+--|-----------------|---
  4    9    TOTALS| 3  8 13 12 10                              | 46
(?=help) > M 2.2

Modify assignment to yard INGALLS
(Blanks denote no change; use zero for assignment deletion)
      Period:| 1  2  3  4  5  6  7  8  9
   Shipclass T|86 87 88 89 90 91 92 93 94
  2 CG-47      | 2  2  2  2
               .. .. .. .. .. .. .. .. ..
                              2

(?=help) > M 2.3

Modify assignment to yard INGALLS
(Blanks denote no change; use zero for assignment deletion)
      Period:| 1  2  3  4  5  6  7  8  9
   Shipclass T|86 87 88 89 90 91 92 93 94
  3 DDG-51     |   F1  2  2  2
               .. .. .. .. .. .. .. .. ..
                        3  3  3

(?=help) > A 2

Make new assignment for yard # 2, INGALLS
      Period:| 1  2  3  4  5  6  7  8  9
   Shipclass T|86 87 88 89 90 91 92 93 94
   ........... .. .. .. .. .. .. .. .. ..
   DD-963

Warning: you have assigned no ships of this new class
(?=help) > &
```

The new class name appears but with 0 assignments, as one would expect. JOHN proceeds to fix his minor mistake using the Modify command, adding awards for five DD-963's. He gives more DDG's to TODD LA and TODD SEA.

```
Scenario: DEMO                 *SHIP ASSIGNMENTS*   Page  1A   Time in: FISCYR
Yard     Period:| 1  2  3  4  5  6  7  8  9                            | 9
   Shipclass T  |86 87 88 89 90 91 92 93 94                            |TOT
BIW          #01|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 CG-47       | 1  2  1  2  1                                        |  7
  2 DDG-51      |   Y1 F2  1  2                                        |  6
INGALLS      #02|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 BB-61      r|    1                                                 |  1
  2 CG-47       | 2  2  2  2  2                                        | 10
  3 DD-963      |                                                      |  0
  4 DDG-51      |   F1  3  3  3                                        | 10
  5 LHD         |      L2  1  1                                        |  4
TODD LA      #03|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 DDG-51      |   Y1  2  2  2                                        |  7
TODD SEA     #04|--+--+--+--+--+--+--+--+--|---------------------------|---
  1 AFDM        |       1     1                                        |  2
  2 DDG-51      |      Y1 F2  1                                        |  4
                |--+--+--+--+--+--+--+--+--|---------------------------|---
  4  10   TOTALS| 3  8 14 13 13                                        | 51
(?=help) > M 2.3

Modify assignment to yard INGALLS
(Blanks denote no change; use zero for assignment deletion)
         Period:| 1  2  3  4  5  6  7  8  9
     Shipclass T|86 87 88 89 90 91 92 93 94
  3 DD-963      |
                 .. .. .. .. .. .. .. .. ..
                   1  1  1  1  1

(?=help) > M 3.1

Modify assignment to yard TODD LA
(Blanks denote no change; use zero for assignment deletion)
         Period:| 1  2  3  4  5  6  7  8  9
     Shipclass T|86 87 88 89 90 91 92 93 94
  1 DDG-51      |   Y1  2  2  2
                 .. .. .. .. .. .. .. .. ..
                             3  3

(?=help) > M 4.2

Modify assignment to yard TODD SEA
(Blanks denote no change; use zero for assignment deletion)
         Period:| 1  2  3  4  5  6  7  8  9
     Shipclass T|86 87 88 89 90 91 92 93 94
  2 DDG-51      |      Y1 F2  1
                 .. .. .. .. .. .. .. .. ..
                       Y2    3

(?=help) > &
```

After reviewing his changes JOHN decides he is done. But as he looks at the assignments for INGALLS, noticing the LHDs, he is reminded that he forgot to check out that missing lead ship job description for LSD-49's that cropped up during the last assigner run. Even though there are no LSD-49's loaded during this run, he decides to go take care of the matter right away before he forgets again.

You can put the assigner on "hold" while you go do other things, much as the DBU is left on hold whenever you leave it. To exit the assigner via a "hold" JOHN gives the "H" command. He is back in the assigner choice menu, where he pops to menu TOP.

```
Scenario: DEMO              *SHIP ASSIGNMENTS*   Page  1A   Time in: FISCYR
Yard      Period:| 1  2  3  4  5  6  7  8  9                           |  9
  Shipclass     T|86 87 88 89 90 91 92 93 94                           |TOT
BIW           #01|--+--+--+--+--+--+--+--+--|--------------------------|---
 1 CG-47         | 1  2  1  2  1                                       |  7
 2 DDG-51        |   Y1 F2  1  2                                       |  6
INGALLS       #02|--+--+--+--+--+--+--+--+--|--------------------------|---
 1 BB-61        r|    1                                                |  1
 2 CG-47         | 2  2  2  2  2                                       | 10
 3 DD-963        | 1  1  1  1  1                                       |  5
 4 DDG-51        |   F1  3  3  3                                       | 10
 5 LHD           |      L2  1  1                                       |  4
TODD LA       #03|--+--+--+--+--+--+--+--+--|--------------------------|---
 1 DDG-51        |   Y1  2  3  3                                       |  9
TODD SEA      #04|--+--+--+--+--+--+--+--+--|--------------------------|---
 1 AFDM          |       1     1                                       |  2
 2 DDG-51        |       2 F2  3                                       |  7
                 |--+--+--+--+--+--+--+--+--|--------------------------|---
 4  10     TOTALS| 4  9 16 15 17                                       | 61
(?=help) > H
```

//

```
Menu is ASSIGN           * ALIAS COMMAND SYSTEM *    Scenario is DEMO
------------------------------------------------------------------------
                     MANUAL ASSIGNER SPECIFICATIONS
   1.  ASSIGNER INITIALIZATION PARAMETERS
   2.  EXECUTE THE ASSIGNER

 COMMAND:/
```

There he asks for the DBU; the MASTER screen comes up quickly since the DBU itself was on hold. JOHN knows that construction schedule job descriptions are handled in the NC_SKED_PF (new construction schedule planning factors) screen, so he asks for it immediately using the "=" command.

```
Menu is TOP                  * ALIAS COMMAND SYSTEM *    Scenario is DEMO
_____________________________________________________________________________
                          TOP LEVEL ALIAS COMMAND MENU
   1.  CUSTOMIZE USER ENVIRONMENT
   2.  CALL NON-ALIAS PROCESSORS
   3.  DATA BASE UPDATING SYSTEM
   4.  MANUAL ASSIGNMENT EDITOR
   5.  FORCE LEVEL REPORT GENERATOR
   6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 3
```

Back in the DBU now.

```
//////////////////////////////////////////////////////////////////////////

SCREEN IS: MASTER                                   SCENARIO IS: DEMO
=============================================================================
 ?  for help            ALIAS DATA BASE UPDATE SYSTEM              =NAME jumps
================choose a screen to use by number or >NAME=====================

                        COMMAND: =NC_SKED_PP

                        1.  SHIP CLASSES
                        2.  SHIP JOB TYPES
                        3.  SHIP JOB SCHEDULES
                        4.  SHIP YARDS
                        5.  SHIP DEACTIVATIONS
                        6.  DATA DICTIONARY UPDATING SYSTEM



=========================No data may be changed here==========================
    Please place a command or option number after COMAMND and press RETURN
```

He does a K (clear, not shown), and then a Search for LSD-49 job descriptions. The job description for the ORDFOL (ordinary follow) series type is retrieved.

```
SCREEN IS: NC_SKED_PF               LATEST DATA          SCENARIO IS: DEMO
==========================================================================
 ? for help               ALIAS DATA BASE UPDATE SYSTEM          =NAME jumps
========================Schedule Planning Factors=========================

                             COMMAND: S
              job                                    intervals
    Class:                   LSD-49            Approp-Award:
    Job Type:                                  Award-Start:
    Yard:                                      Start-Keel:
    Commissioning:                             Keel-Launch:
    Sequence Type:                             Launch_Delivery:
    Construction Method:                       Deliv-Commission:
    Customer:                                  in Time Units:
    Complexity group:                          Default Award Day:
    Days Added to
    Service Life                                        Data Source
                                                        Data Date
                                                        Entry Date
                                                        Entry By
==Your privileges in this screen are: inspect, add, modify, update, delete==
                Place a command after COMMAND and press RETURN
```

```
//////////////////////////////////////////////////////////////////////
```

```
SCREEN IS: NC_SKED_PF               LATEST DATA          SCENARIO IS: DEMO
==========================================================================
 ? for help               ALIAS DATA BASE UPDATE SYSTEM          =NAME jumps
========================Schedule Planning Factors=========================

                             COMMAND:
              job                                    intervals
    Class:                   LSD-49            Approp-Award:      1
    Job Type:                NEWCON            Award-Start:       12
    Yard:                    ANY               Start-Keel:        8
    Commissioning:           1                 Keel-Launch:       23
    Sequence Type:           ORDFOL            Launch_Delivery:   24
    Construction Method:     MODULZ            Deliv-Commission:  1
    Customer:                USN               in Time Units:     MONTHS
    Complexity group:                          Default Award Day: 11/01
    Days Added to            0
    Service Life                                        Data Source 908
                                                        Data Date   8/01/1984
                                                        Entry Date  8/02/1984
                                                        Entry By    DBA
==Your privileges in this screen are: inspect, add, modify, update, delete==
                Place a command after COMMAND and press RETURN
```

After inspecting the job description, he decides that it is not appropriate for the LEAD job and makes a number of changes (shown in `bold`). Then he Adds ("A" command) the new record to the data base. Notice that the Entry Date and Entry By fields reflect the current date and JOHN's user name after the Add is complete.

Having made his change, JOHN exits the DBU by giving the "Q" command.

```
SCREEN IS: NC_SKED_PF            LATEST DATA          SCENARIO IS: DEMO
===========================================================================
? for help            ALIAS DATA BASE UPDATE SYSTEM                =NAME jumps
=========================Schedule Planning Factors=========================

                              COMMAND: A
              job                                      intervals
    Class:                    LSD-49          Approp-Award:          1
    Job Type:                 NEWCON          Award-Start:           16
    Yard:                     ANY             Start-Keel:            10
    Commissioning:            1               Keel-Launch:           24
    Sequence Type:            LEAD            Launch_Delivery:       26
    Construction Method:      MODULZ          Deliv-Commission:      1
    Customer:                 USN             in Time Units:         MONTHS
    Complexity group:                         Default Award Day:     11/01
    Days Added to             0
    Service Life
                                                     Data Source IMAGINED
                                                     Data Date   10/16/1984
                                                     Entry Date  8/02/1984
                                                     Entry By    DBA
===Your privileges in this screen are: inspect, add, modify, update, delete===
                    Place a command after COMMAND and press RETURN
```

//

```
SCREEN IS: NC_SKED_PF            LATEST DATA          SCENARIO IS: DEMO
===========================================================================
? for help            ALIAS DATA BASE UPDATE SYSTEM                =NAME jumps
=========================Schedule Planning Factors=========================

                              COMMAND: Q
              job                                      intervals
    Class:                    LSD-49          Approp-Award:          1
    Job Type:                 NEWCON          Award-Start:           16
    Yard:                     ANY             Start-Keel:            10
    Commissioning:            1               Keel-Launch:           24
    Sequence Type:            LEAD            Launch_Delivery:       26
    Construction Method:      MODULZ          Deliv-Commission:      1
    Customer:                 USN             in Time Units:         MONTHS
    Complexity group:                         Default Award Day:     11/01
    Days Added to             0
    Service Life
                                                     Data Source IMAGINED
                                                     Data Date   10/16/1984
                                                     Entry Date  10/26/1984
                                                     Entry By    JOHNA
===Your privileges in this screen are: inspect, add, modify, update, delete===
                    Place a command after COMMAND and press RETURN
```

Returned to the TOP menu, he moves again into the assigner choice menu and chooses option 2, "execute the assigner". Since the assigner was left on hold, this does not re-start it---JOHN is given a message telling him he is back in the assigner and then the assigner command prompt. He is effectively right where he was when he decided to jump over to the DBU. Since he was finished in the assigner, he gives the "Q" command to trigger the schedule creation and update step.

```
Menu is TOP                * ALIAS COMMAND SYSTEM *   Scenario is DEMO
-----------------------------------------------------------------------
                     TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 4
```

///

```
Menu is ASSIGN             * ALIAS COMMAND SYSTEM *   Scenario is DEMO
-----------------------------------------------------------------------
                     MANUAL ASSIGNER SPECIFICATIONS
    1.  ASSIGNER INITIALIZATION PARAMETERS
    2.  EXECUTE THE ASSIGNER

  COMMAND: 2

YOU HAVE RETURNED TO AN ACTIVE ASSIGNER SESSION

          UNDER SCENARIO DEMO

IF THIS IS NOT YOUR CURRENT SCENARIO, YOU MUST LEAVE
THE ASSIGNER WITH THE "Q" COMMAND AND THEN RE-RUN IT.
Press RETURN to continue:

(?=help) > Q
```

The usual prompt and progress messages appear. If you are following along on your terminal, notice that schedule creation and update for 60 ships in four yards is significantly faster than four 160 ships in a dozen or so yards.

```
The assigner will now update ship schedules for the current scenario.

Only projected ship schedules will be updated.  Any changes made
during this session which imply changes to historical or current
job schedules will be ignored.

If you made NO CHANGES, or if you want your changes DISCARDED,
considerable time can be saved by skipping this update.  If you
do skip it, any changes you have made will be lost.

Do you want to skip the update? N

Opening data base files for update.
Updating projected ships data base.
Updating schedules for yard BIW
Updating schedules for yard INGALLS
Updating schedules for yard TODD LA
Updating schedules for yard TODD SEA
Updating hull numbers in schedules.
Update complete. Display buffers purged. Closing DB.
```

Returned to the assigner choice menu after the update, JOHN pops to the TOP Command System menu and asks for "Non-ALIAS Processors".

```
Menu is ASSIGN              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
                        MANUAL ASSIGNER SPECIFICATIONS
     1.  ASSIGNER INITIALIZATION PARAMETERS
     2.  EXECUTE THE ASSIGNER

  COMMAND: /
```

//

```
Menu is TOP                 * ALIAS COMMAND SYSTEM *    Scenario is DEMO
--------------------------------------------------------------------------
                         TOP LEVEL ALIAS COMMAND MENU
     1.  CUSTOMIZE USER ENVIRONMENT
     2.  CALL NON-ALIAS PROCESSORS
     3.  DATA BASE UPDATING SYSTEM
     4.  MANUAL ASSIGNMENT EDITOR
     5.  FORCE LEVEL REPORT GENERATOR
     6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 2
```

These are useful programs which run on the HP. JOHN's purpose now is to examine the format control file he has used to generate his battle group reports. He must use a text editor to do this, so he asks for the best one (the Text-Document Processor (TDP)) by choosing menu option 2. TDP comes up just as it would if the "TDP" command were given in response to an MPE ":" prompt.

JOHN uses the "T" (Text) command to load in the format control file (called BGPOM86.FMTFIL), and then asks for a listing of all the lines in the file.

We will note a few things in passing about format control files, although a thorough description is left for later in this manual. Lines which start with a "%" are "comment" lines---they are ignored by the report generator, and thus may be included as documentation or for readability.

Notice that the title which has appeared on the reports appears on the TITLE lines in the file.

```
Menu is HPROGS            * ALIAS COMMAND SYSTEM *   Scenario is DEMO
--------------------------------------------------------------------------
                                HP PROGRAMS
     1.  EDITOR
     2.  TDP
     3.  GRAPH
     4.  RELATE
     5.  MENU
     6.  SPOOK

  COMMAND: 2
TDP/3000     (A.03.08)  HP36578 Editor (c) COPYRIGHT Hewlett-Packard Co. 1983
             MON, OCT 29, 1984, 12:05 AM  (DAY #303)
/T BGPOM86.FMTFIL
/L ALL

        % ALIAS BATTLE GROUP REPORT FORMAT/CONTENTS DEFINITION FILE
        % format is: title; start; type; function; bgroup; makeup; end
        % title line has titles for report
        % start line indicates start of processing
        % type line indicates ship classes making up a type
        % function line lists types which can perform a function, in
        %  order of preference
        % bgroup  describes battle groups to be made up
        % makeup  describes which functions each battle group requires
        %
        TITLE Deployable Battle Group ProjectiFor POM-86
        TITLE Based on Surface Combatant Requirements Only
        TITLE (All Data Notional)
        %
        START
        %
        % type format similar to force level report: name,label,class list
```

A battle group report format control file is basically divided into four blocks or sections: TYPE, FUNCTION, BGROUP, and MAKEUP. In the TYPE lines ship classes are aggregated into ship types, each with a single name and a label for use in the BALANCE section of the report. The FUNCTION lines define functional categories of ships, each having a list of TYPEs capable of performing the function (in descending order of suitability). Note that in line 127 it is specified that BB's can substitute for CRUISERs, for example.

The BGROUP lines define the battle groups which are desired and the priority and target number of each. For example, in line 135 CVBG'are given a priority of 1 and a target number of 15, with the target effective from a long time ago to a long time in the future (it is possible to have to target number change over time). The label that will appear on the output for this group is "CARRIER BG".

The MAKEUP lines define what ships are required to make up one unit of a group, in terms of numbers of FUNCTIONal categories. In line 142 of the file a CVBG is specified to require a carrier, a cruiser, 4 DDG's, 2 DD's, and 4 frigates.

In looking over the file JOHN decides three things. The first is that CRUISERs and BBs should be able to substitute for DDGs. On the last report he noticed a relative surplus of CRUISERs in the near term, which could relieve some of the DDG shortage. Editing implements this decision. Second, he decides that 8 DD's is too many to require for a Marine Amphibious Force, and changes the requirement to only 4. The third is that it is perhaps not too much of a surprise that he did not achieve 17 carrier battle groups, since the target was only 15! He uses TDP command to change the target to 17.

He keeps the file under a new name (which leaves the original version intact, unchanged), BGPOM86B.FMTFIL, and exits the editor.

To find out more about using the TDP editor, see Hewlett-Packard's documentation for their standard editor EDITOR or for TDP itself. There is an introductory manual for EDITOR that is likely to be particularly helpful.

```
116      %
117      TYPE CARRIER,CARRIER,  CV-41,CV-59,CV-63,CV-67,CVN-65,CVN-68
118      TYPE BB,BB,            BB-61
119      TYPE CRUISER,CRUISER,  CGN-25,CGN-36,CGN-38,CGN-35,
119.1    +                    CGN-9,CG-16,CG-26,CG-47
120      TYPE DDG,DDG,            DDG-2,DDG-37,DDG-51,DDG-993
121      TYPE DD,DD            DD-945,DD-963
122      TYPE FFG,FFG,               FFG-1,FFG-7
123      TYPE FF,FF              FF-1037,FF-1040,FF-1052
124      %
125      % function format is name,list of types which can perform it
126      %  in order of preference
127      FUNCTION CRUISER, CRUISER,BB
128      FUNCTION CARRIER, CARRIER
129      FUNCTION DDG,     DDG
130      FUNCTION DD,      DD
131      FUNCTION FRIGATE, FFG,FF
132      %
133      % bgroup format is name,output label,priority,target level,
134      %  begin date this defn takes effect, end date this defn effective
135      BGROUP CVBG,CARRIER BG,1,15,1/1/1900,1/1/2111
136      BGROUP SAG,SURFACE AG, 3, 4,1/1/1900,1/1/2111
137      BGROUP MAF,MARINE AF,  2, 2,1/1/1900,1/1/2111
138      BGROUP ESC,SUPPLY ESCORT,4,10,1/1/1900,1/1/2111
139      BGROUP CON,CONVOY,     5,10,1/1/1900,1/1/2111
140      %
141      % makeup format is battle group name, function, # reqd, func, #reqd
142      MAKEUP CVBG,   CARRIER,1,CRUISER,1,DDG,4,DD,2,FRIGATE,4
143      MAKEUP SAG,    CRUISER,2,DDG,2,FRIGATE,2
144      MAKEUP MAF,    CRUISER,2,DDG,2,DD,8,FRIGATE,10
145      MAKEUP ESC,    DDG,1,DD,1,FRIGATE,2
146      MAKEUP CON,    DD,1,FRIGATE,4
147      %
148      STOP

/M 129
  129    FUNCTION DDG,     DDG
  Changes:                    R,CRUISER,BB
  129    FUNCTION DDG,     DDG,CRUISER,BB
  Changes:
/M 144
  144    MAKEUP MAF,    CRUISER,2,DDG,2,DD,8,FRIGATE,10
  Changes:                                 R4
  144    MAKEUP MAF,    CRUISER,2,DDG,2,DD,4,FRIGATE,10
  Changes:
/M 135
  135    BGROUP CVBG,CARRIER BG,1,15,1/1/1900,1/1/2111
  Changes:                         R7
  135    BGROUP CVBG,CARRIER BG,1,17,1/1/1900,1/1/2111
  Changes:
/K BGPOM86B.FMTFIL
/E
```

Back in the Command System again, JOHN pops to the TOP and heads for the force level report generators again.

```
Menu is HPROGS              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
------------------------------------------------------------------------
                                HP PROGRAMS
     1.  EDITOR
     2.  TDP
     3.  GRAPH
     4.  RELATE
     5.  MENU
     6.  SPOOK

  COMMAND: /
```

//

```
Menu is TOP                 * ALIAS COMMAND SYSTEM *    Scenario is DEMO
------------------------------------------------------------------------
                         TOP LEVEL ALIAS COMMAND MENU
     1.  CUSTOMIZE USER ENVIRONMENT
     2.  CALL NON-ALIAS PROCESSORS
     3.  DATA BASE UPDATING SYSTEM
     4.  MANUAL ASSIGNMENT EDITOR
     5.  FORCE LEVEL REPORT GENERATOR
     6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND: 5
```

He executes the battle group report generator, giving it the name of the modified format control file he just created. The usual progress messages result.

```
Menu is FLRPTG              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
---------------------------------------------------------------------------
                          FORCE LEVEL GENERATOR
     1.  FORCE LEVEL REPORT INITIALIZATION PARAMETERS
     2.  EXECUTE FORCE REPORT GENERATOR
     3.  PREPARE BATTLE GROUP REPORT

  COMMAND: 3

Starting up Battle Group Report Generator. Please stand by.
NAME OF OUTPUT CONTROL FILE (name.group, or ^)BGPOM86B.FMTFIL

The file FLREPT already exists in your group.
The force level modules usually stores its report in that
file when you request that the report be saved on disk.
Since that file is in use, what file would you like to store
the report in? (name <= 8 letters, or / to not save report): BGREPTB


//////////////////////////////////////////////////////////////////////


Opening required data base files...
Looking for ships in data base...
Looking for class BB-61
Looking for class CG-16
Looking for class CG-26
Looking for class CG-47
Looking for class CGN-25
Looking for class CGN-35
Looking for class CGN-36
Looking for class CGN-38
Looking for class CGN-9
Looking for class CV-41
Looking for class CV-59
Looking for class CV-63
Looking for class CV-67
Looking for class CVN-65
Looking for class CVN-68
Looking for class DD-963
Looking for class DDG-2
Looking for class DDG-37
Looking for class DDG-51
Looking for class DDG-993
Looking for class FF-1040
Looking for class FF-1052
Looking for class FFG-1
Looking for class FFG-7
```

The output is beginning to look satisfactory. Given JOHN's revised POM, 17 carrier battle groups will be deployable starting in 1997. The uniformly small number of ships "left over" as specified in the BALANCE section of the report indicates that JOHN's final guesses concerning the numbers of additional ships required were quite good.

It may seem odd to you that the report implies that the Navy will be unable to deploy significant numbers of <u>any</u> of the other kinds of battle groups (SAGs, MAF, etc.). Remember that this report is based on notional data only, though. It is not meant to be realistic (security would forbid publishing a realistic projection in this unclassified manual in any case); it is meant only as an illustration of the general way in which ALIAS capabilities can be used.

An inspection of the current and historical construction schedules being used by scenario DEMO (which actually belong to FIXIT, since JOHN is using FIXIT's schedules in these epochs indirectly) would show that a significant number of cruisers and destroyers currently available are not included. Also, the names of some classes were omitted from the class lists in the format control file.

Now provisionally satisfied with the report's contents, JOHN is ready to get a copy of the schedules produced by the assigner for his POM (for review) and an estimate of the POM's yearly cost.

ALIAS currently has no full-fledged cost-estimation module. However, JOHN knows it is possible to use the capabilities of the RELATE query language to prepare a rough cost estimate. He can print out the schedules using RELATE as well.

Since RELATE must be run using his "R" user name, the next step is to leave ALIAS, which he does.

DEPLOYABLE BATTLE GROUP PROJECTIFOR POM-86
BASED ON SURFACE COMBATANT REQUIREMENTS ONLY
(ALL DATA NOTIONAL)

CALENDAR YEAR	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
BATTLEGROUP														
CARRIER BG	10	10	10	11	11	11	12	13	16	16	16	17	17	17
SURFACE AG														
MARINE AF			1					1		1	1			
SUPPLY ESCORT	3	3				4	2	1	2			2	2	2
CONVOY	8	8	7	10	10	9	10	6	2					
BALANCE														
CARRIER	3	3	3	3	3	4	2	2						
BB														
1 1														
CRUISER														
DDG										4	4			
DD					1									
FFG														
FF	33	33	34	28	28	26	22	26	35	33	31	26	21	16

//

Menu is FLRPTG * ALIAS COMMAND SYSTEM * Scenario is DEMO

FORCE LEVEL GENERATOR

1. FORCE LEVEL REPORT INITIALIZATION PARAMETERS
2. EXECUTE FORCE REPORT GENERATOR
3. PREPARE BATTLE GROUP REPORT

COMMAND: Q

Sure you want to terminate your ALIAS session? Y

END OF PROGRAM

He logs on under his "R" user name.

```
:HELLO JOHNR.SEA90
ENTER USER PASSWORD:

HP3000 / MPE IV C.B1.A2.  SAT, OCT 20, 1984,  2:40 PM
**************************************************************
WEEKLY BACK UP OF FILES IS NOW TAKING 8- 2400 FEET REELS
OF TAPE AND LASTING MORE THAN 2 1/2 HOURS.  IT IS IMPERATIVE
THAT USERS OF THE SYSTEM PURGE OLD FILES ON A REGULAR BASIS.
**************************************************************


Bulletin:
**********************************************************************
**********************************************************************
****************** DO NOT LEAVE YOUR TERMINAL ***********************
******************        UNATTENDED          ***********************
******************     SIGN    OFF    !!!!     ***********************
**********************************************************************
**********************************************************************
```

END OF PROGRAM

Once in RELATE, JOHN turns first to preparation of the schedule printout. He opens the ncjodat.projj file, which holds the projected schedules for all scenarios. Then he asks for a list of the indexes which exist on the file---basically, these are his choices for the sort order in which the schedules will print out. He chooses the first one (via the SET INDEX 1) command so that the schedules will be printed by yard and class, the same sorting he is used to seeing on an assigner page.

```
:RELATE
Comment  MAKE SURE HP2934A IS TURNED ON BEFORE PRinting,
Comment  OTHERWISE YOU'LL LOSE SPOOLER OWNERSHIP.
FILE rdbecat.pub.sys=rdbecat.pub.relate
FILE rdbhelp.pub.sys=rdbhelp.pub.relate
FILE  grecat.pub.sys= grecat.pub.relate
FILE plotter;dev=50
FILE rdblist;dev=58;cctl
RUN   relate.pub.relate;lib=p;maxdata=31000

RELATE/3000 V4.40A MON, OCT 29, 1984, 12:27 AM (C) CRI

  1)OPEN FILE NCJODAT.PROJJ;MODE=SHARED
  2)SHOW INDEXES

FILE NAME              =NCJODAT.PROJJ.SEA90

INDEX 1 BY SCENARIO,YARD,CLASS,NCJOBT,HULL,COMNUM
  KEY LENGTH (IN WORDS): 23
  DISTRIBUTION         : 159,13,4,2,1,1
  NODE SIZE (IN WORDS) : 1
  KEYS PER BLOCK       : 22
  LEVELS IN TREE       : 2
  NUMBER OF USED NODES : 20
  STORAGE UTILIZATION  : 78%

INDEX 2 BY SCENARIO,CLASS,HULL,COMNUM;UNARY
  KEY LENGTH (IN WORDS): 16
  DISTRIBUTION         : 159,6,1,1
  NODE SIZE (IN WORDS) : 1
  KEYS PER BLOCK       : 30
  LEVELS IN TREE       : 2
  NUMBER OF USED NODES : 16
  STORAGE UTILIZATION  : 68%

INDEX 3 BY SCENARIO,CLASS,DELIVERY
  KEY LENGTH (IN WORDS): 16
  DISTRIBUTION         : 159,6,1
  NODE SIZE (IN WORDS) : 1
  KEYS PER BLOCK       : 30
  LEVELS IN TREE       : 2
  NUMBER OF USED NODES : 14
  STORAGE UTILIZATION  : 77%

  3)SET INDEX 1
INDEX #1 IS NOW THE CURRENT INDEX.
```

JOHN then wants to know what fields are in the file. He gives the SHOW command and RELATE lists them. He picks out the fields he is interested in seeing for each schedule and puts them in a PRINT command. Since he is mainly interested in dates, he asks for only the fields required to identify each ship-job and for the milestone date fields (omitting the commission milestone). Notice that the first word of the command reads PR:S:P. "PR" is shorthand for "PRINT". The ":S" tells RELATE to print the fields in the same column order as he has them named in the PR command. The ":P" tells it to do the print on the SEA 90 dot matrix printer instead of on his terminal. Thus he will have hard copy to take back to his desk.

The output is shown in Figure 3-1. Notice that JOHN's run of the assigner for only four yards did not destroy the schedules for all other yards---they were ignored or bypassed by the assigner during all phases of its operation. Notice also that some of the schedules have blanks for some dates. Did the assigner make an error? No, those schedules are for fiscal 1985 ships, and the assigner runs were for fiscal 1986 and beyond. Those schedules were therefore untouched by all the assigner runs; the blank dates were inherited from FIXIT. Notice that all hull numbers are ordered by delivery by the assigner (independent of yard). Notice how the assigner spread the start dates of ships within a class and yard relatively evenly (for schedules it generated). Take the three LSD-41's awarded to Avondale in fiscal 1986 and 1987. Their start dates are about 8 months apart (3 ships started over a 24 month period=8 month intervals), even though two of them are awarded on the same day.

4) SHOW

FILE NAME =NCJODAT.PROJJ.SEA90

NAME	T Y P	PRINT LEN	INT SIZE
SCENARIO	A	12	12B
CLASS	A	10	10B
HULL	I	4	1W
COMNUM	I	6	1W
YARD	A	8	8B
NCJOBT	A	6	6B
JSTYP	A	6	6B
CUSTOMER	A	8	8B
SHIPNAME	A	30	30B
CMETHD	A	6	6B
APPROP	D	10	2W
AWARD	D	10	2W
START	D	10	2W
KEEL	D	10	2W
LAUNCH	D	10	2W
DELIVERY	D	10	2W
COMMISSION	D	10	2W
DAYSADDED	I	9	1W
ASNORDER	D	8	2W
DATADATE	D	10	2W
DATASOURCE	A	10	10B
ENTRY_BY	A	8	8B
ENTRY_DATE	D	10	2W
AUTOMOD	A	4	4B
PROGVAR1	I	8	1W
PROGVAR2	I	8	1W
SUBRELUMAP	D	10	2W

PRINT LINE WIDTH = 283 CHARACTERS.

5) **PR:S:P**
YARD,CLASS,HULL,NCJOBT,JSTYP,APPROP,AWARD,START,KEEL,LAUNCH,DELIVERY

THE OUTPUT HAS BEEN PLACED IN SPOOL FILE #1013

Figure 3-1. Sample Schedules

YARD	CLASS	HULL	NCJOBT	JSTYP	APPROP	AWARD	START	KEEL	LAUNCH	DELIVERY
AVONDALE	LSD-41	41	NEWCON	ORDFOL	10/01/1984	11/30/1984	1/31/1986			8/30/1988
AVONDALE	LSD-41	42	NEWCON	ORDFOL	10/01/1984	11/30/1984	6/30/1986			1/30/1989
AVONDALE	LSD-41	43	NEWCON	ORDFOL	10/01/1985	11/01/1985	11/01/1986	5/01/1987	1/01/1989	5/01/1990
AVONDALE	LSD-41	44	NEWCON	ORDFOL	10/01/1985	11/01/1985	7/02/1987	1/02/1988	9/02/1989	1/02/1991
AVONDALE	LSD-41	45	NEWCON	ORDFOL	10/01/1986	11/01/1986	3/01/1988	9/01/1988	5/01/1990	9/01/1991
AVONDALE	LSD-49	49	NEWCON	LEAD	10/01/1987	11/01/1987	11/01/1988	7/01/1989	6/01/1991	6/01/1993
AVONDALE	LSD-49	50	NEWCON	ORDFOL	10/01/1987	11/01/1987	5/02/1989	1/02/1990	12/02/1991	12/02/1993
AVONDALE	LSD-49	51	NEWCON	ORDFOL	10/01/1988	11/01/1988	11/01/1989	7/01/1990	6/01/1992	6/01/1994
AVONDALE	LSD-49	52	NEWCON	ORDFOL	10/01/1988	11/01/1988	5/03/1990	1/03/1991	12/03/1992	12/03/1994
AVONDALE	LSD-49	53	NEWCON	ORDFOL	10/01/1989	11/01/1989	11/01/1990	7/01/1991	6/01/1993	6/01/1995
AVONDALE	LSD-49	54	NEWCON	ORDFOL	10/01/1989	11/01/1989	5/02/1991	1/02/1992	12/02/1993	12/02/1995
AVONDALE	T-AO-187	187	NEWCON	ORDFOL	10/01/1985	11/01/1985	12/01/1985	1/01/1986	11/01/1986	11/01/1987
AVONDALE	T-AO-187	188	NEWCON	ORDFOL	10/01/1984	11/30/1984	1/30/1986			4/30/1988
AVONDALE	T-AO-187	189	NEWCON	ORDFOL	10/01/1985	11/01/1985	6/01/1986	7/01/1986	5/01/1987	5/01/1988
AVONDALE	T-AO-187	190	NEWCON	ORDFOL	10/01/1984	11/30/1984	5/30/1986			8/31/1988
AVONDALE	T-AO-187	191	NEWCON	ORDFOL	10/01/1986	11/01/1986	12/01/1986	1/01/1987	11/01/1987	11/01/1988
AVONDALE	T-AO-187	192	NEWCON	ORDFOL	10/01/1984	11/30/1984	9/30/1986			12/30/1988
AVONDALE	T-AO-187	193	NEWCON	ORDFOL	10/01/1986	11/01/1986	6/01/1987	7/01/1987	5/01/1988	5/01/1989
AVONDALE	T-AO-187	194	NEWCON	ORDFOL	10/01/1987	11/01/1987	12/01/1987	1/01/1988	11/01/1988	11/01/1989
AVONDALE	T-AO-187	195	NEWCON	ORDFOL	10/01/1987	11/01/1987	5/31/1989	6/30/1988	4/30/1989	4/30/1990
AVONDALE	T-AO-187	196	NEWCON	ORDFOL	10/01/1988	11/01/1988	12/01/1988	1/01/1989	11/01/1989	11/01/1990
AVONDALE	T-AO-187	197	NEWCON	ORDFOL	10/01/1988	11/01/1988	6/01/1989	7/01/1989	5/01/1990	5/01/1991
AVONDALE	T-AO-187	198	NEWCON	ORDFOL	10/01/1989	11/01/1989	12/01/1989	1/01/1990	11/01/1990	11/01/1991
AVONDALE	T-AO-187	199	NEWCON	ORDFOL	10/01/1989	11/01/1989	6/01/1990	7/01/1990	5/01/1991	5/01/1992
BIW	CG-47	61	NEWCON	ORDFOL	10/01/1984	11/30/1984	8/30/1986			10/31/1989
BIW	CG-47	63	NEWCON	ORDFOL	10/01/1985	11/01/1985	1/01/1987	7/01/1987	9/01/1988	5/01/1990
BIW	CG-47	66	NEWCON	ORDFOL	10/01/1986	11/01/1986	1/01/1988	7/01/1988	9/01/1989	5/01/1991
BIW	CG-47	69	NEWCON	ORDFOL	10/01/1986	11/01/1986	9/18/1988	3/18/1989	5/18/1990	1/18/1992
BIW	CG-47	71	NEWCON	ORDFOL	10/01/1987	11/01/1987	6/06/1989	12/06/1989	2/06/1991	10/06/1992
BIW	CG-47	74	NEWCON	ORDFOL	10/01/1988	11/01/1988	2/21/1990	8/21/1990	10/21/1991	6/21/1993
BIW	CG-47	76	NEWCON	ORDFOL	10/01/1988	11/01/1988	11/09/1990	5/09/1991	7/09/1992	3/09/1994
BIW	CG-47	79	NEWCON	ORDFOL	10/01/1989	11/01/1989	7/28/1991	1/28/1992	3/28/1993	11/28/1994
BIW	DDG-51	53	NEWCON	YOLEAD	10/01/1986	11/01/1986	2/01/1988	11/01/1988	1/01/1990	10/01/1991
BIW	DDG-51	56	NEWCON	1STFOL	10/01/1987	11/01/1987	11/01/1988	8/01/1989	10/01/1990	7/01/1992
BIW	DDG-51	62	NEWCON	ORDFOL	10/01/1987	11/01/1987	6/17/1989	3/17/1990	5/17/1991	2/17/1993
BIW	DDG-51	66	NEWCON	ORDFOL	10/01/1988	11/01/1988	1/31/1990	10/31/1990	12/31/1991	9/30/1993
BIW	DDG-51	73	NEWCON	ORDFOL	10/01/1989	11/01/1989	11/01/1990	8/01/1991	10/01/1992	7/01/1994
BIW	DDG-51	78	NEWCON	ORDFOL	10/01/1989	11/01/1989	6/17/1991	3/17/1992	5/17/1993	2/17/1995
EB GROT	SSBN-726	736	NEWCON	ORDFOL	10/01/1984	12/30/1984	1/30/1985			12/30/1990
EB GROT	SSBN-726	737	NEWCON	ORDFOL	10/01/1985	11/01/1985	12/01/1985	1/01/1987	2/01/1990	9/01/1991
EB GROT	SSBN-726	738	NEWCON	ORDFOL	10/01/1986	11/01/1986	12/01/1986	1/01/1988	2/01/1991	9/01/1992
EB GROT	SSBN-726	739	NEWCON	ORDFOL	10/01/1987	11/01/1987	12/01/1987	1/01/1989	2/01/1992	9/01/1993
EB GROT	SSBN-726	740	NEWCON	ORDFOL	10/01/1988	11/01/1988	12/01/1988	1/01/1990	2/01/1993	9/01/1994
EB GROT	SSBN-726	741	NEWCON	ORDFOL	10/01/1989	11/01/1989	12/01/1989	1/01/1991	2/01/1994	9/01/1995
EB GROT	SSN-21	21	NEWCON	LEAD	10/01/1988	11/01/1988	11/01/1988	11/01/1989	5/01/1991	9/01/1993
EB GROT	SSN-688	757	NEWCON	ORDFOL	10/01/1984	11/30/1984	5/31/1985			12/31/1989
EB GROT	SSN-688	759	NEWCON	ORDFOL	10/01/1984	11/30/1984	11/30/1985			5/30/1990

Figure 3-1. Sample Schedules (Cont'd).

EB GROT	SSN-688	759	NEWCON	ORDFOL	10/01/1985	11/01/1985	11/01/1986	1/01/1988	12/01/1989	3/01/1991
EB GROT	SSN-688	761	NEWCON	ORDFOL	10/01/1985	11/01/1985	5/22/1987	7/22/1988	6/22/1990	9/22/1991
EB GROT	SSN-688	763	NEWCON	ORDFOL	10/01/1986	11/01/1986	12/11/1987	2/11/1989	1/11/1991	4/11/1992
EB GROT	SSN-688	765	NEWCON	ORDFOL	10/01/1986	11/01/1986	7/01/1988	9/01/1989	8/01/1991	11/01/1992
EB GROT	SSN-688	767	NEWCON	ORDFOL	10/01/1987	11/01/1987	1/20/1989	3/20/1990	2/20/1992	5/20/1993
EB GROT	SSN-688	769	NEWCON	ORDFOL	10/01/1987	11/01/1987	8/10/1989	10/10/1990	9/10/1992	12/10/1993
EB GROT	SSN-688	771	NEWCON	ORDFOL	10/01/1988	11/01/1988	3/01/1990	5/01/1991	4/01/1993	7/01/1994
EB GROT	SSN-688	773	NEWCON	ORDFOL	10/01/1989	11/01/1989	11/01/1990	1/01/1992	12/01/1993	3/01/1995
EB GROT	SSN-688	776	NEWCON	ORDFOL	10/01/1989	11/01/1989	5/23/1991	7/23/1992	6/23/1994	9/23/1995
GDQ	AE	1	NEWCON	YDLEAD	10/31/1987	11/01/1987	11/16/1987	11/22/1987	12/03/1987	12/15/1987
GDQ	AE	2	NEWCON	ORDFOL	10/31/1988	11/01/1988	11/16/1988	11/22/1988	12/03/1988	12/15/1988
GDQ	AE	3	NEWCON	ORDFOL	10/31/1989	11/01/1989	11/16/1989	11/22/1989	12/03/1989	12/15/1989
GDQ	AG	1	NEWCON	LEAD	10/31/1985	11/01/1985	11/16/1985	11/22/1985	12/03/1985	12/15/1985
GDQ	AO-187	191	CONV	ORDFOL	10/01/1987	11/01/1987	12/01/1987	1/01/1988	11/01/1988	11/01/1989
GDQ	AO-187	192	CONV	ORDFOL	10/01/1988	11/01/1988	12/01/1988	1/01/1989	11/01/1989	11/01/1990
GDQ	T-AG	1	NEWCON	LEAD	10/01/1984	4/30/1985	10/30/1985			10/30/1987
GDQ	T-AG	2	NEWCON	1STFOL	10/01/1984	4/30/1985	2/28/1986			2/28/1988
INGALLS	BB-61	61	REACT	ORDFOL	10/01/1986	11/01/1986	12/01/1986	12/01/1986	12/01/1986	8/01/1988
INGALLS	CG-47	60	NEWCON	ORDFOL	10/01/1984	11/30/1984	4/30/1986			6/30/1989
INGALLS	CG-47	62	NEWCON	ORDFOL	10/01/1984	11/30/1984	10/31/1986			12/30/1989
INGALLS	CG-47	64	NEWCON	ORDFOL	10/01/1985	11/01/1985	1/01/1987	7/01/1987	9/01/1988	5/01/1990
INGALLS	CG-47	65	NEWCON	ORDFOL	10/01/1985	11/01/1985	7/02/1987	1/02/1988	3/02/1989	11/02/1990
INGALLS	CG-47	67	NEWCON	ORDFOL	10/01/1986	11/01/1986	1/01/1988	7/01/1988	9/01/1989	5/01/1991
INGALLS	CG-47	68	NEWCON	ORDFOL	10/01/1986	11/01/1986	7/01/1988	1/01/1989	3/01/1990	11/01/1991
INGALLS	CG-47	70	NEWCON	ORDFOL	10/01/1987	11/01/1987	1/01/1989	7/01/1989	9/01/1990	5/01/1992
INGALLS	CG-47	72	NEWCON	ORDFOL	10/01/1987	11/01/1987	7/02/1989	1/02/1990	3/02/1991	11/02/1992
INGALLS	CG-47	73	NEWCON	ORDFOL	10/01/1988	11/01/1988	1/01/1990	7/01/1990	9/01/1991	5/01/1993
INGALLS	CG-47	75	NEWCON	ORDFOL	10/01/1988	11/01/1988	7/02/1990	1/02/1991	3/02/1992	11/02/1993
INGALLS	CG-47	77	NEWCON	ORDFOL	10/01/1989	11/01/1989	1/01/1991	7/01/1991	9/01/1992	5/01/1994
INGALLS	CG-47	78	NEWCON	ORDFOL	10/01/1989	11/01/1989	7/02/1991	1/02/1992	3/02/1993	11/02/1994
INGALLS	DD-963	998	NEWCON	ORDFOL	10/01/1985	11/01/1985	5/01/1986	1/01/1987	11/01/1987	3/01/1989
INGALLS	DD-963	999	NEWCON	ORDFOL	10/01/1986	11/01/1986	5/01/1987	1/01/1988	11/01/1988	3/01/1990
INGALLS	DD-963	1000	NEWCON	ORDFOL	10/01/1987	11/01/1987	5/01/1988	1/01/1989	11/01/1989	3/01/1991
INGALLS	DD-963	1001	NEWCON	ORDFOL	10/01/1988	11/01/1988	5/01/1989	1/01/1990	11/01/1990	3/01/1992
INGALLS	DD-963	1002	NEWCON	ORDFOL	10/01/1989	11/01/1989	5/01/1990	1/01/1991	11/01/1991	3/01/1993
INGALLS	DDG-51	51	NEWCON	LEAD	10/01/1984	12/30/1984	4/30/1986			6/30/1989
INGALLS	DDG-51	52	NEWCON	1STFOL	10/01/1986	11/01/1986	11/01/1987	8/01/1988	10/01/1989	7/01/1991
INGALLS	DDG-51	57	NEWCON	ORDFOL	10/01/1987	11/01/1987	11/01/1988	8/01/1989	10/01/1990	7/01/1992
INGALLS	DDG-51	59	NEWCON	ORDFOL	10/01/1987	11/01/1987	3/27/1989	12/27/1989	2/27/1991	11/27/1992
INGALLS	DDG-51	63	NEWCON	ORDFOL	10/01/1987	11/01/1987	8/20/1989	5/20/1990	7/20/1991	4/20/1993
INGALLS	DDG-51	67	NEWCON	ORDFOL	10/01/1988	11/01/1988	3/03/1990	12/03/1990	2/03/1992	11/03/1993
INGALLS	DDG-51	70	NEWCON	ORDFOL	10/01/1988	11/01/1988	7/02/1990	4/02/1991	6/02/1992	3/02/1994
INGALLS	DDG-51	72	NEWCON	ORDFOL	10/01/1988	11/01/1988	10/31/1990	7/31/1991	9/30/1992	6/30/1994
INGALLS	DDG-51	76	NEWCON	ORDFOL	10/01/1989	11/01/1989	3/03/1991	12/03/1991	2/03/1993	11/03/1994
INGALLS	DDG-51	79	NEWCON	ORDFOL	10/01/1989	11/01/1989	7/02/1991	4/02/1992	6/02/1993	3/02/1995
INGALLS	DDG-51	82	NEWCON	ORDFOL	10/01/1989	11/01/1989	10/31/1991	7/31/1992	9/30/1993	6/30/1995
INGALLS	LHD	1	NEWCON	LEAD	10/01/1987	11/01/1987	2/01/1989	4/01/1990	6/01/1992	11/01/1993
INGALLS	LHD	2	NEWCON	ORDFOL	10/01/1987	11/01/1987	11/01/1989	1/01/1991	3/01/1993	9/01/1994
INGALLS	LHD	3	NEWCON	ORDFOL	10/01/1988	11/01/1988	9/02/1990	10/02/1991	12/02/1993	5/02/1995
INGALLS	LHD	4	NEWCON	ORDFOL	10/01/1989	11/01/1989	5/02/1991	7/02/1992	9/02/1994	2/02/1996

Figure 3-1. Sample Schedules (Cont'd).

MARINET	MCM-1	1	NEWCON	ORDFOL	10/01/1984	11/30/1984	12/30/1984			5/30/1987
MARINET	MCM-1	3	NEWCON	ORDFOL	10/01/1984	11/30/1984	4/30/1985			9/30/1987
MARINET	MCM-1	5	NEWCON	ORDFOL	10/01/1985	11/01/1985	11/01/1986	4/01/1987	4/01/1988	5/01/1989
MARINET	MCM-1	8	NEWCON	ORDFOL	10/01/1985	11/01/1985	7/02/1987	12/02/1987	12/02/1988	1/02/1990
MARINET	MCM-1	9	NEWCON	ORDFOL	10/01/1986	11/01/1986	3/01/1988	8/01/1988	8/01/1989	9/01/1990
MARINET	MSH-1	1	NEWCON	LEAD	10/01/1984	10/15/1984	8/30/1985			8/30/1987
MARINET	MSH-1	2	NEWCON	1STFOL	10/01/1985	11/01/1985	8/01/1986	12/01/1986	11/01/1987	8/01/1988
MARINET	MSH-1	3	NEWCON	ORDFOL	10/01/1985	11/01/1985	10/31/1986	2/28/1987	1/31/1988	10/31/1988
MARINET	MSH-1	4	NEWCON	ORDFOL	10/01/1985	11/01/1985	1/30/1987	5/30/1987	4/30/1988	1/30/1989
MARINET	MSH-1	5	NEWCON	ORDFOL	10/01/1985	11/01/1985	5/01/1987	9/01/1987	8/01/1988	5/01/1989
MARINET	MSH-1	6	NEWCON	ORDFOL	10/01/1986	11/01/1986	8/01/1987	12/01/1987	11/01/1988	8/01/1989
MARINET	MSH-1	7	NEWCON	ORDFOL	10/01/1986	11/01/1986	10/31/1987	2/29/1988	1/31/1989	10/31/1989
MARINET	MSH-1	8	NEWCON	ORDFOL	10/01/1986	11/01/1986	1/30/1988	5/30/1988	4/30/1989	1/30/1990
MARINET	MSH-1	9	NEWCON	ORDFOL	10/01/1986	11/01/1986	4/30/1988	8/30/1988	7/30/1989	4/30/1990
MARINET	MSH-1	10	NEWCON	ORDFOL	10/01/1987	11/01/1987	9/01/1988	12/01/1988	11/01/1989	8/01/1990
MARINET	MSH-1	11	NEWCON	ORDFOL	10/01/1987	11/01/1987	10/31/1988	2/28/1989	1/31/1990	10/31/1990
MARINET	MSH-1	12	NEWCON	ORDFOL	10/01/1987	11/01/1987	1/30/1989	5/30/1989	4/30/1990	1/30/1991
MARINET	MSH-1	13	NEWCON	ORDFOL	10/01/1987	11/01/1987	5/01/1989	9/01/1989	8/01/1990	5/01/1991
MARINET	MSH-1	14	NEWCON	ORDFOL	10/01/1988	11/01/1988	8/01/1989	12/01/1989	11/01/1990	8/01/1991
MARINET	MSH-1	15	NEWCON	ORDFOL	10/01/1988	11/01/1988	10/31/1989	2/28/1990	1/31/1991	10/31/1991
MARINET	MSH-1	16	NEWCON	ORDFOL	10/01/1988	11/01/1988	1/30/1990	5/30/1990	4/30/1991	1/30/1992
MARINET	MSH-1	17	NEWCON	ORDFOL	10/01/1988	11/01/1988	5/01/1990	9/01/1990	8/01/1991	5/01/1992
NASSCO	AO-187	193	CONV	ORDFOL	10/01/1988	11/01/1988	12/01/1988	1/01/1989	11/01/1989	11/01/1990
NASSCO	AO-187	194	CONV	ORDFOL	10/01/1989	11/01/1989	12/01/1989	1/01/1990	11/01/1990	11/01/1991
NASSCO	AO-187	195	CONV	ORDFOL	10/01/1989	11/01/1989	8/01/1990	9/01/1990	7/01/1991	7/01/1992
NASSCO	AOE	1	NEWCON	LEAD	10/01/1986	11/01/1986	2/01/1988	8/01/1988	7/01/1989	8/01/1990
NASSCO	AOE	2	NEWCON	1STFOL	10/01/1987	11/01/1987	2/01/1989	8/01/1989	7/01/1990	8/01/1991
NASSCO	AOE	3	NEWCON	ORDFOL	10/01/1988	11/01/1988	2/01/1990	8/01/1990	7/01/1991	8/01/1992
NASSCO	AOE	4	NEWCON	ORDFOL	10/01/1989	11/01/1989	2/01/1991	8/01/1991	7/01/1992	8/01/1993
NASSCO	AR	1	NEWCON	LEAD	10/01/1989	11/01/1989	2/01/1991	8/01/1991	7/01/1992	8/01/1993
NASSCO	LPD-4	16	CONV	ORDFOL	10/01/1987	11/01/1987	2/01/1989	4/01/1990	6/01/1992	11/01/1993
NASSCO	LPD-4	17	CONV	ORDFOL	10/01/1988	11/01/1988	2/01/1990	4/01/1991	6/01/1993	11/01/1994
NASSCO	LPD-4	20	CONV	ORDFOL	10/01/1989	11/01/1989	2/01/1991	4/01/1992	6/01/1994	11/01/1995
NASSCO	LPD-4	22	CONV	ORDFOL	10/01/1989	11/01/1989	11/01/1991	1/01/1993	3/01/1995	8/01/1996
NNEWS	CVN-68	74	NEWCON	ORDFOL	10/01/1985	11/01/1985	11/01/1986	12/01/1987	4/01/1991	1/01/1993
NNEWS	CVN-68	75	NEWCON	ORDFOL	10/01/1987	11/01/1987	11/01/1988	12/01/1989	4/01/1993	1/01/1995
NNEWS	MTS	1	CONV	ORDFOL	10/01/1986	11/01/1986	12/01/1986	1/01/1987	11/01/1987	11/01/1988
NNEWS	SSN-688	756	NEWCON	ORDFOL	10/01/1984	11/30/1984	4/30/1985			10/30/1989
NNEWS	SSN-688	760	NEWCON	ORDFOL	10/01/1985	11/01/1985	11/01/1986	1/01/1988	12/01/1999	3/01/1991
NNEWS	SSN-688	762	NEWCON	ORDFOL	10/01/1985	11/01/1985	5/22/1987	7/22/1988	6/22/1990	9/22/1991
NNEWS	SSN-688	764	NEWCON	ORDFOL	10/01/1986	11/01/1986	12/11/1987	2/11/1989	1/11/1991	4/11/1992
NNEWS	SSN-688	766	NEWCON	ORDFOL	10/01/1986	11/01/1986	7/01/1988	9/01/1989	8/01/1991	11/01/1992
NNEWS	SSN-688	768	NEWCON	ORDFOL	10/01/1987	11/01/1997	1/20/1989	3/20/1990	2/20/1992	5/20/1993
NNEWS	SSN-688	770	NEWCON	ORDFOL	10/01/1987	11/01/1987	8/10/1989	10/10/1990	9/10/1992	12/10/1993
NNEWS	SSN-688	772	NEWCON	ORDFOL	10/01/1988	11/01/1988	3/01/1990	5/01/1991	4/01/1993	7/01/1994
NNEWS	SSN-688	774	NEWCON	ORDFOL	10/01/1989	11/01/1989	11/01/1990	1/01/1992	12/01/1993	3/01/1995
NNEWS	SSN-688	775	NEWCON	ORDFOL	10/01/1989	11/01/1989	5/23/1991	7/23/1992	6/23/1994	9/23/1995
PENNSHIP	LPD-4	18	CONV	ORDFOL	10/01/1988	11/01/1988	2/01/1990	4/01/1991	6/01/1993	11/01/1994
PENNSHIP	LPD-4	19	CONV	ORDFOL	10/01/1988	11/01/1988	10/02/1990	12/02/1991	2/02/1994	7/02/1995
PENNSHIP	LPD-4	21	CONV	ORDFOL	10/01/1989	11/01/1989	6/02/1991	8/02/1992	10/02/1994	3/02/1996

Figure 3-1. Sample Schedules (Cont'd).

PENNSHIP	T-ACS	1	CONV	ORDFOL	10/01/1984	11/30/1984	11/30/1984			4/30/1986
PENNSHIP	T-ACS	2	CONV	ORDFOL	10/01/1984	11/30/1984	11/30/1984			5/30/1986
PENNSHIP	T-ACS	3	CONV	ORDFOL	10/01/1985	11/01/1985	12/01/1985	1/01/1986	11/01/1986	11/01/1987
PENNSHIP	T-ACS	4	CONV	ORDFOL	10/01/1985	11/01/1985	5/06/1986	6/06/1986	4/06/1987	4/06/1988
PENNSHIP	T-ACS	5	CONV	ORDFOL	10/01/1985	11/01/1985	10/09/1986	11/09/1986	9/09/1987	9/09/1988
PENNSHIP	T-ACS	6	CONV	ORDFOL	10/01/1986	11/01/1986	3/14/1987	4/14/1987	2/14/1988	2/14/1989
PENNSHIP	T-ACS	7	CONV	ORDFOL	10/01/1986	11/01/1986	8/18/1987	9/18/1987	7/18/1988	7/18/1989
PENNSHIP	T-ACS	8	CONV	ORDFOL	10/01/1987	11/01/1987	1/21/1988	2/21/1988	12/21/1988	12/21/1989
PENNSHIP	T-ACS	9	CONV	ORDFOL	10/01/1987	11/01/1987	6/25/1988	7/25/1988	5/25/1989	5/25/1990
PENNSHIP	T-AVB	1	CONV	ORDFOL	10/01/1984	11/30/1984	12/31/1984			7/31/1985
PENNSHIP	T-AVB	2	CONV	ORDFOL	10/01/1985	11/01/1985	12/01/1985	1/01/1986	11/01/1986	11/01/1987
PETERSON	MCM-1	2	NEWCON	ORDFOL	10/01/1984	11/30/1984	12/30/1984			6/30/1987
PETERSON	MCM-1	4	NEWCON	ORDFOL	10/01/1984	11/30/1984	5/30/1985			10/30/1987
PETERSON	MCM-1	6	NEWCON	ORDFOL	10/01/1985	11/01/1985	11/01/1986	4/01/1987	4/01/1988	5/01/1989
PETERSON	MCM-1	7	NEWCON	ORDFOL	10/01/1985	11/01/1985	5/02/1987	10/02/1987	10/02/1988	11/02/1989
TACOMA	T-AGOS-1	1	NEWCON	ORDFOL	10/01/1984	11/30/1984	5/30/1986			10/30/1987
TACOMA	T-AGOS-1	2	NEWCON	ORDFOL	10/01/1985	11/01/1985	12/01/1985	1/01/1986	11/01/1986	11/01/1997
TACOMA	T-AGOS-1	3	NEWCON	ORDFOL	10/01/1984	11/30/1984	9/30/1986			2/28/1988
TACOMA	T-AGOS-1	4	NEWCON	ORDFOL	10/01/1985	11/01/1985	4/01/1986	5/01/1986	3/01/1987	3/01/1988
TACOMA	T-AGOS-1	5	NEWCON	ORDFOL	10/01/1984	11/30/1984	1/30/1987			6/30/1988
TACOMA	T-AGOS-1	6	NEWCON	ORDFOL	10/01/1985	11/01/1985	7/31/1986	8/31/1986	6/30/1987	6/30/1988
TODD LA	DDG-51	54	NEWCON	YOLEAD	10/01/1986	11/01/1986	2/01/1988	11/01/1988	1/01/1990	10/01/1991
TODD LA	DDG-51	58	NEWCON	ORDFOL	10/01/1987	11/01/1987	11/01/1988	8/01/1989	10/01/1990	7/01/1992
TODD LA	DDG-51	60	NEWCON	ORDFOL	10/01/1987	11/01/1987	4/02/1989	1/02/1990	3/02/1991	12/02/1992
TODD LA	DDG-51	65	NEWCON	ORDFOL	10/01/1988	11/01/1988	11/01/1989	8/01/1990	10/01/1991	7/01/1993
TODD LA	DDG-51	68	NEWCON	ORDFOL	10/01/1988	11/01/1988	4/02/1990	1/02/1991	3/02/1992	12/02/1993
TODD LA	DDG-51	71	NEWCON	ORDFOL	10/01/1988	11/01/1988	9/01/1990	6/01/1991	8/01/1992	5/01/1994
TODD LA	DDG-51	75	NEWCON	ORDFOL	10/01/1989	11/01/1989	3/03/1991	12/03/1991	2/03/1993	11/03/1994
TODD LA	DDG-51	90	NEWCON	ORDFOL	10/01/1989	11/01/1989	7/02/1991	4/02/1992	6/02/1993	3/02/1995
TODD LA	DDG-51	83	NEWCON	ORDFOL	10/01/1989	11/01/1989	10/31/1991	7/31/1992	9/30/1993	6/30/1995
TODD SEA	AFDM	1	NEWCON	ORDFOL	10/01/1987	11/01/1987	12/01/1987	1/01/1988	11/01/1988	11/01/1989
TODD SEA	AFDM	2	NEWCON	ORDFOL	10/01/1989	11/01/1989	12/01/1989	1/01/1990	11/01/1990	11/01/1991
TODD SEA	DDG-51	55	NEWCON	YOLEAD	10/01/1987	11/01/1987	11/01/1988	8/01/1989	10/01/1990	7/01/1992
TODD SEA	DDG-51	61	NEWCON	ORDFOL	10/01/1987	11/01/1987	4/06/1989	1/06/1990	3/06/1991	12/06/1992
TODD SEA	DDG-51	64	NEWCON	1STFOL	10/01/1988	11/01/1988	11/01/1989	8/01/1990	10/01/1991	7/01/1993
TODD SEA	DDG-51	69	NEWCON	ORDFOL	10/01/1988	11/01/1988	4/06/1990	1/06/1991	3/06/1992	12/06/1993
TODD SEA	DDG-51	74	NEWCON	ORDFOL	10/01/1989	11/01/1989	11/01/1990	8/01/1991	10/01/1992	7/01/1994
TODD SEA	DDG-51	77	NEWCON	ORDFOL	10/01/1989	11/01/1989	4/06/1991	1/06/1992	3/06/1993	12/06/1994
TODD SEA	DDG-51	81	NEWCON	ORDFOL	10/01/1989	11/01/1989	9/09/1991	6/09/1992	8/09/1993	5/09/1995
TODD SEA	FFG-7	61	NEWCON	ORDFOL	10/01/1983	11/30/1984	12/30/1985			9/30/1988

187 LINES PRINTED

Having printed the schedules, JOHN is ready to generate his cost estimate. He will do this by combining the data in the schedule relation and that in a relation which specifies ship unit costs by class, job type, and job series type into a single relation. This relation will have one record for each ship, with the ship's cost included in the record. Once he has this relation, he can ask RELATE to add up the costs for him in various ways.

If you were doing this exercise for the first time, you would need to make up the relation of unit costs and figure out how to ask RELATE to do what you want. However, JOHN has done it before and so already has a unit cost relation. Also, he has saved the RELATE commands which must be given in an editor-type file (called an EXECUTE file in this context).

JOHN first closes the ncjodat.projj relation (with the "CLOSE" command) because he knows the execute file will just open it up again. Then he executes the file, which is stored in costrept.rprocs ("COST REPorT.Relate PROCedure fileS"). By placing the ";SHOW" option on the EXECUTE command, JOHN has asked RELATE to show him the commands in the file as it executes them.

First ncjodat.projj is opened, then a file called unitcst.descj. The latter contains the unit cost estimates which JOHN entered earlier, which are in millions of dollars, and keyed by ship class, job type, and job series type.

Then a SELECT is given which "joins" the schedule and unit cost files using the class, job type, and series type fields, and which restricts the records returned from ncjodat.projj to those for scenario DEMO. The RELATE manuals describe SELECT and its capabilities in detail. For now, think of a join as taking two files, laying them on a table side by side, and glueing them together. A retriction then throws out all the lines which are not of interest (i.e. which don't have identical key values appearing in both files).

The selection asks that only three of the many fields in the two files be returned (printable) for each ship job: ship class, year of appropriation, and the unit cost for the particular job.

The result is copied to a new relation named TMP. One drawback of RELATE is that only one selection at a time may be given. Since JOHN is going to need to give another select "on top of" this one, he must put the results of this one into a file so the new select may be given on the data in that file. The file will have one record for each projected ship construction/ conversion/ reactivation job in scenario DEMO, with the name of the class the job is being done on and the cost of the job.

```
 6)CLOSE
 7)EXECUTE COSTREPT.RPROCS;SHOW
 8)NOTE  PRODUCES A CRUDE COST REPORT USING UNIT COSTS
 9)NOTE  AS SPECIFIED IN UNITCST.DESCJ
10)NOTE
11)OPEN FILE NCJODAT.PROJJ;PATH=P;MODE=SHARED
12)OPEN FILE UNITCST.DESCJ;PATH=C;MODE=SHARED
13)SELECT CLASS,YEAR=$YEAR(P.APPROP),C.COST WHERE &
.1&)P.CLASS=C.CLASS AND P.NCJOBT=C.NCJOBT AND P.JSTYP=C.JSTYP &
.2&)AND P.SCENARIO="DEMO"
14)COPY TO TMP
THE "TMP" FILE HAS BEEN CREATED AS A PERMANENT RELATE/3000 FILE.
187 LINES COPIED.
15)OPEN FILE TMP
16)SELECT YEAR,TOTCOST=$SUM(COST BY YEAR) UNIQUE BY YEAR
WARNING: TEMPORARY INDEX WILL BE CREATED TO SATISFY AGGREGATE BY CLAUSE.
17)PRINT

YEAR TOTCOS

1983    435
1984  13050
1985  18135
1986  14572
1987  24415
1988  19470
1989  21110

7 LINES PRINTED.

18)SELECT YEAR,CLASS,TOTCOST=$SUM(COST BY YEAR,CLASS) UNIQUE BY YEAR,CLASS
WARNING: TEMPORARY INDEX WILL BE CREATED TO SATISFY AGGREGATE BY CLAUSE.
19)PRINT:P

THE OUTPUT HAS BEEN PLACED IN SPOOL FILE #1014

20)PURGE FILE TMP
THE "TMP" FILE HAS BEEN PURGED.
```

Now it is only necessary to sum up the costs. The next selection does this, lumping jobs together by year of appropriation and summing up all the costs within that year. The "UNIQUE BY YEAR" clause ensures that the cost for each year is only printed once. JOHN prints the resulting estimate to his display.

He was hoping for a fairly even distribution of costs over the period, but finds an uneven one. He will need to revise his schedule of awards somewhat so that extraordinary appropriations are not required in any given year. To support this revision he will need to know the breakdown of yearly costs in more detail--- he wants costs by year and class.

All the necessary information is already in the TMP file. He gives a new SELECT which is a minor modification of the one given in the execute file (he adds "CLASS" to the aggregate BY clause), and asks for a print to the SEA 90 dot matrix printer. The result is shown in Figure 3-2. He then purges the TMP file since he is done with it.

Figure 3-2. Sample Costs by Year and Class

YEAR	CLASS	TOTCOS
1983	FFG-7	435
1984	CG-47	3150
1984	DDG-51	1005
1984	LSD-41	960
1984	MCM-1	2980
1984	MSH-1	60
1984	SSBN-726	1850
1984	SSN-688	2100
1984	T-ACS	120
1984	T-AG	265
1984	T-AGOS-1	240
1984	T-AO-187	300
1984	T-AVB	20
1985	AG	560
1985	CG-47	3150
1985	CVN-68	4560
1985	DD-963	500
1985	LSD-41	960
1985	MCM-1	2980
1985	MSH-1	135
1985	SSBN-726	1850
1985	SSN-688	2800
1985	T-ACS	180
1985	T-AGOS-1	240
1985	T-AO-187	200
1985	T-AVB	20
1986	AOE	750
1986	BB-61	232
1986	CG-47	4200
1986	DD-963	500
1986	DDG-51	2525
1986	LSD-41	480
1986	MCM-1	745
1986	MSH-1	120
1986	MTS	50
1986	SSBN-726	1850
1986	SSN-688	2800
1986	T-ACS	120
1986	T-AO-187	200
1987	AE	650
1987	AFDM	200
1987	AO-187	75
1987	AOE	650
1987	CG-47	3150
1987	CVN-68	4560
1987	DD-963	500
1987	DDG-51	7295
1987	LHD	1060
1987	LPD-4	35
1987	LSD-49	1150
1987	MSH-1	120
1987	SSBN-726	1850
1987	SSN-688	2800
1987	T-ACS	120
1987	T-AO-187	200
1988	AE	600
1988	AO-187	150
1988	AOE	600
1988	CG-47	4200
1988	DD-963	500
1988	DDG-51	7295
1988	LHD	500
1988	LPD-4	105
1988	LSD-49	1100
1988	MSH-1	120
1988	SSBN-726	1850
1988	SSN-21	850
1988	SSN-688	1400
1988	T-AO-187	200
1989	AE	600
1989	AFDM	200
1989	AO-187	150
1989	AOE	600
1989	AR	500
1989	CG-47	3150
1989	DD-963	500
1989	DDG-51	8855
1989	LHD	500
1989	LPD-4	105
1989	LSD-49	1100
1989	SSBN-726	1850
1989	SSN-688	2800
1989	T-AO-187	200

JOHN decides he would also like a printout of the unit costs which his estimates are based on, so he does a SET PATH C to gain access to the unitcst.descj file (which was opened with a path name of "C"), looks to see what fields are in it, and prints its contents to the dot matrix printer (see Table 3-3).

Finished with his session, JOHN leaves RELATE and logs off the computer.

We want to emphasize the power and usefulness of the method that JOHN just used to get his cost estimate. The use of "job description" or "unit cost" or "labor requirements" relations in combination with the schedules in the ALIAS data base can support conduct of many analyses in addition to those that ALIAS already supports. To see how to undertake these analyses, spend some time exploring the capabilities of RELATE, and in particular the use of aggregates in the SELECT command.

```
 21) SET PATH C
 22) SHOW

FILE NAME              =UNITCST.DESCJ.SEA90

              T
              Y PRINT  INT
NAME          P  LEN   SIZE

CLASS         A  10    10B
NCJCBT        A   6     6B
JSTYP         A   6     6B
COST          I   6     1W

PRINT LINE WIDTH = 35 CHARACTERS.
 33) PRINT:P

THE OUTPUT HAS BEEN PLACED IN SPOOL FILE #1016
 34) //

END OF PROGRAM
:BYE
```

A lot of things have been introduced in this sample session. Among them are:

1) How to log on to the ALIAS host computer
2) How to run the ALIAS Core and its Command System
3) How to do ad hoc queries of the ALIAS data base, and how to combine these capabilities with Core capabilities to analyze ship acquisition programs
4) What scenarios are and how to create and work with them
5) The assigner, which makes creation of program schedules easy
6) The battle group report generator, which lets you assess the impact of a program on the force structure
7) The Data Base Updating system, which lets you inspect and change the data base

In a more general sense, you have seen most of the contexts which you are likely to encounter when working with ALIAS, and you have seen most of ALIAS' major capabilities.

It may seem at this point that there is a bewildering array of things that you have to know. As a system of building blocks or tools, it is true that ALIAS offers you many options and ways of doing things. However, there is guidance for most steps in the form of menus, and it is often not necessary to know all the options.

Not shown during the sample session is the extensive array of on-line help which ALIAS makes available. You can ask for this help at any time by giving the "?" command. It is a good idea to give this as your first command whenever you are doing something you have never done before, just to find out what kinds of help are available in case you get confused.

The most important subtle thing to remember about ALIAS is that nearly everything uses the contents of the data base in some way. You must understand what data is in a scenario when you begin to use it, and keep track of the changes you make to it as you go along. When strange results appear for no seeming good reason, think about the data elements the results are based on and what pattern or element values might yield the results. This is what JOHN did when he got in trouble in the sample session; and notice that most of the times he got in trouble were a result of his not being very familiar with his starting point, the FIXIT scenario.

The remaining Sections of this manual will describe in more detail how to operation each of ALIAS's constituent parts.

Figure 3-3. Sample Unit Costs

$LINE	CLASS	NCJOBT	JSTYP	COST
1	AE	NEWCON	ORDFOL	600
2	AE	NEWCON	YDLEAD	650
3	AFDM	NEWCON	ORDFOL	200
4	AG	NEWCON	LEAD	560
5	AO-187	CONV	ORDFOL	75
6	AOE	NEWCON	1STFOL	650
7	AOE	NEWCON	LEAD	750
8	AOE	NEWCON	ORDFOL	600
9	AR	NEWCON	LEAD	500
10	BB-61	REACT	ORDFOL	232
11	CG-47	NEWCON	ORDFOL	1050
12	CVN-68	NEWCON	ORDFOL	4560
13	DD-963	NEWCON	ORDFOL	500
14	DDG-51	NEWCON	1STFOL	855
15	DDG-51	NEWCON	LEAD	1005
16	DDG-51	NEWCON	ORDFOL	805
17	DDG-51	NEWCON	YDLEAD	835
18	FFG-7	NEWCON	ORDFOL	435
19	LHD	NEWCON	LEAD	560
20	LHD	NEWCON	ORDFOL	500
21	LPD-4	CONV	ORDFOL	35
22	LSD-41	NEWCON	ORDFOL	480
23	LSD-49	NEWCON	LEAD	600
24	LSD-49	NEWCON	ORDFOL	550
25	MCM-1	NEWCON	ORDFOL	745
26	MSH-1	NEWCON	1STFOL	45
27	MSH-1	NEWCON	LEAD	60
	H-1	NEWCON	ORDFOL	30
			ORDFOL	50
30	SSBN-726	NEWCON	ORDFOL	1850
31	SSN-21	NEWCON	LEAD	850
32	SSN-688	NEWCON	ORDFOL	700
33	T-ACS	CONV	ORDFOL	60
34	T-AG	NEWCON	1STFOL	125
35	T-AG	NEWCON	LEAD	140
36	T-AGOS-1	NEWCON	ORDFOL	80
37	T-AO-187	NEWCON	ORDFOL	100
38	T-AVB	CONV	ORDFOL	20

38 LINES PRINTED.

4.0 USING THE ALIAS COMMAND SYSTEM

The Command System is the framework on which ALIAS analytical capabilities are hung. It shows you lists of the things which you can do, accepts your choices, and ensures that your wishes are carried out. It is menu-oriented, meaning that ALIAS options are shown in related groups, with only one such group being shown at a time.

To use the Command system you need only know a few commands and conventions, all of which were illustrated during the sample session. There are several additional services and advanced capabilities as well, which were not shown in the sample. This Section is a complete guide to the nature and use of the Command System.

4.1 EXECUTING THE COMMAND SYSTEM

Before you can use the Command System two things must be done. First, you must be given an "A" user name by an ALIAS system administrator. Most people in the ALIAS community have two user names which they can use to log onto the HP, e.g. "JOHNA" and "JOHNR". The "A" name is for running the Command System and all its associated capabilities, while the "R" name is for making direct queries of the data base using RELATE. Security constraints made the 2-name system necessary.

The second requirement is that the ALIAS system administrator set up security such that you will be accepted by ALIAS, i.e., you must be given privileges. Three kinds of privileges will typically be required: basic RELATE usage privileges for both your user names, Command System access privileges, and Data Base Updating system usage privileges. The administrator will give you the privileges just after establishing your user names, so you probably need not worry about them.

Once you are set up you need only log on under your "A" name (sit down at a terminal, hit the return key, and type "HELLO JOHNA.SEA90", using your name instead of "JOHNA"), and give the command "ALIAS" in response to MPE's colon prompt.

4.2 THE NATURE OF COMMANDS

You control what actions ALIAS takes by giving commands to the Command System. As explained in Section 2, there are two basic means of giving commands in ALIAS as a whole: line-oriented and fill-in-the-blank. The Command System expects line-oriented commands exclusively.

Line-oriented systems type out a prompt when they are ready to process your next request; they expect you to type something in response (right after the prompt, using no arrow keys, though use of the backspace key is permitted) and hit the RETURN key. The Command System's prompt is "COMMAND:"; when this is the last line appearing on the display the system is waiting to do your bidding.

The responses you give to the "COMMAND:" prompt will take three basic forms: numbers, other characters, and settings.

The command system will always present you with a menu of choices before giving the prompt; when you type a number in response to the prompt you are indicating that you want to choose that numbered option from the menu. The system will respond by doing whatever the option implies---perhaps it will execute a module, or show you a different menu, or just change something's status.

In addition to choosing from the menu there are "house-keeping" functions which you can perform (generally from any menu), such as ending your session. You invoke one of these functions by giving the proper non-numeric character ("Q" in the case of ending the session).

Settings change the status of a system control variable, such as the start date of interest for a run of a particular module. Settings are typically multi-character commands, in which you indicate which variable you want to change and what you want its new value to be. You indicate the variable by its number on the menu, and its value by typing the value after an equal sign; an example would be "3=10/1/1995".

4.3 TYPES OF COMMAND SYSTEM MENU

The Command System will present you with four types of menu. Each type has a different purpose, and often a somewhat different set of commands which you can give. The types are choice, parameter, list, and help menus.

4.3.1 Choice Menus

Choice menus are action-oriented. If you choose any of the numbered items which appear on a choice menu, the system will take one of two actions: it will show you a different menu, or it will execute a processor. In the latter case you will temporarily leave the Command System and be placed "in" the processor; you will return to the Command System when the processor finishes or you are finished with it.

Figure 4-1 shows the "TOP" Command System choice menu, the one you see first after giving the "ALIAS" command after logging on. The menu has six choices, five of which lead to display of another menu. Command System menus are organized in an hierarchical "tree", as shown in Figure 4-2. You move down the tree by choosing options on successive choice menus, with the options becoming more and more specific as you move down. Notice that the "TOP" menu is the top of the hierarchy in Figure 4-2, and that there are five "branches", one for every option on the "TOP" menu except 3, which causes the Data Base Updating system to be run.

Figure 4-1. Command System Top Menu

```
Menu is TOP                * ALIAS COMMAND SYSTEM *   Scenario is DEMO
--------------------------------------------------------------------------
                           TOP LEVEL ALIAS COMMAND MENU
     1.  CUSTOMIZE USER ENVIRONMENT
     2.  CALL NON-ALIAS PROCESSORS
     3.  DATA BASE UPDATING SYSTEM
     4.  MANUAL ASSIGNMENT EDITOR
     5.  FORCE LEVEL REPORT GENERATOR
     6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:
```

Figure 4-2. Map of Command System Menus

```
                         *** ALIAS MENU SYSTEM MAP ***
                                 ..............
c = choice menu                  : TOP CHOICE :
p = parameter menu               :    MENU    :
l = list menu                    :............:
                                       |
       |----------------|--------------|---------------|----------------|
.......c.......  ......c.....  .......c......  .....c.....  ........c.......
:     USER    :  : OTHER HP :  :  SCENARIO  :  : ASSIGNER :  : FORCE LEVELS :
: ENVIRONMENT :  : PROGRAMS :  :  CONTROL   :  :          :  :              :
:.............:  :..........:  :............:  :..........:  :..............:
       |                             ---------------|                |
.......p.......                      |                        .......p.......
: ENVIRONMENT :                ......p......                  : FORCE LEVEL :
: PARAMETERS  :                :  ASSIGNER  :                 : PARAMETERS  :
:.............:                : PARAMETERS :                 :.............:
                               :............:                        |
                     |---------------|----------------|       .......l.......
                 .....l.....  ......l.......  .....l......    : REPAIR JOBS :
                 : YARDS TO :  : CLASSES TO :  : JOB TYPES :   : CAUSING OUT-:
                 :  ASSIGN  :  :   ASSIGN   :  : TO ASSIGN :   : OF FORCE    :
                 :..........:  :............:  :...........:   :.............:
```

Choice menus are where you get things done in the Command System. If you do not see the exact thing you want to do listed on you current menu, either pick the option which is in the right general area, or go back to the previous menu. The system map shown in Figure 4-2 is also available on-line at any time (give the "?*" command).

There are a few things to notice about the form of choice menus. First, the "* ALIAS COMMAND SYSTEM *" header will always appear at the top of the display in the middle. If you do not see this header, you are not in the Command System. The name of the current menu will be given at the top left, and the name of the scenario you are using at the top right. The title appearing just below the line of dashes should give you a general idea of the subject the menu's options are concerned with.

Figure 4-3 shows an alternative version of the "TOP" menu, one which includes reminders of the most commonly used commands that are available. You decide which version of choice menus will appear by making a choice in the User Environment Parameters menu (more on this in Section 4-5).

Be aware that the Command System is a changing, growing entity. As the number of functions which ALIAS supports grows, the number of menus and the options on each one will grow.

4.3.2 Parameter Menus

A sample parameter menu is shown in Figure 4-4. These menus allow you to make permanent changes to parameters, which are variables controlling how some aspect of the system works. Typically the parameters are read by the module they are associated with and used during the module's set-up and operation. The assigner reads those shown in Figure 4-4 to find out how long a period the assignments table it constructs should cover, etc.

Figure 4-3. Top Menu With Command Tickler Displayed

```
Menu is TOP                 * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu                | /   pop to top menu
    reprint with current values          | {+  build command file
 }   end command file building           |  {   use command  file
 ?   use help processor                  |
_______________________________________________________________________
                     TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:
```

Figure 4-4. Sample Parameter Menu

```
Menu is ASNPRM              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
--------------------------------------------------------------------------
                  MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                = FISCYR       (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE            = 10/ 1/1985   (MM/DD/YYYY)
 3.  ENDING DATE              =  9/31/1994   (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS     = LIST         (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES   = LIST         (ALL/LIST)
 6.  CANDIDATE JOB TYPES      = LIST         (ALL/LIST)
 7.  DISPLAY BASIS            = AWD          (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS             = START        (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE              = PROGRAM      (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION        = PROJ         (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER     = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER      = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH             = OFF          (ON,OFF)

  COMMAND:
```

Parameter menus were created to save you the burden of answering a series of tedious questions each time you want to run a processor (e.g. "What start date?", "What end date?", "What time units?") while still giving you the freedom to make exact specifications of how the processor should work. Since the parameter values are saved permanently as part of the scenario you are working with, you need set them only once (but you can make changes to these settings whenever you want).

Thus the primary commands you will give in a parameter menu will be to change parameter values. Such commands take the form "#=new_value", where # is the number of the parameter you want to change (as shown at the left of the menu). For example, to implement the STARTING DATE setting shown in Figure 4-4, you would give the command "2=10/1/1985". Only one such command can appear on a given line, i.e. only one may be given in response to each "COMMAND:" prompt.

There are six kinds of parameter, and thus six formats in which new values can be specified:

1) CODE WORDS: exemplified by parameters 1 and 7-13 in Figure 4-4, these character-type parameters have a list of valid values, which is given in parentheses following the current value. In changing the setting you must pick from the list.

2) DATES: exemplified by parameters 2 and 3 in Figure 4-4, these are calendar dates specified in MM/DD/YYYY format, i.e. with a 4-digit century. If you specify "1/1/88" by accident the system will take you at your word and think you mean the year A.D. 88.

3) YES/NO: not shown on the menu in the Figure, these parameters can take on a YES or NO (or TRUE or FALSE) value.

4) INTEGERS: not shown in the sample menu, these parameters must have integer number values within the range shown in the parentheses.

5) REAL NUMBERS: like integers, but decimal points are allowed.

6) LIST menu gates: exemplified by parameters 4 through 6 in Figure 4-4, these parameters are concerned with lists of items which can have on-off statuses. As shown in the example (in the parentheses), such parameters can take on values of "ALL" or of "LIST". If ALL, all member of the represented list are "turned on". If LIST, you must look at the list menu to determine which ones are on. To cause the list menu to be shown, just re-set the parameter value to LIST with the command "#=LIST", even if LIST is already the current value.

The format of parameter menus is similar to that of choice menus in that the top line reminds you that you are using the Command System and gives the name of the menu and scenario you are working with. However, the numbered menu options are split into three parts. On the left, following its number, each option has a label briefly explaining what the parameter is for. The current value is shown in the middle, following the "=" sign, while lists of valid values or hints about value formats appear at the right in parentheses.

4.3.3 List Menus

The first page of the list represented by the CANDIDATE SHIP YARDS parameter on the menu shown in Figure 4-4 is shown in Figure 4-5. Lists are really just a special, multi-valued type of parameter which you use to indicate the things you want processing done for. The assigner uses this particular list, which contains the names of all the shipyards that the DEMO scenario has data for, to determine which shipyards it should load and update assignments for.

List menus are the means of making choices from lists. You make choices by turning some or all of the list members (called candidates) "on". Those that are turned on have asterisks by their names. To change a candidate's status, just give its number in response to the "COMMAND:" prompt. If it was off it will be turned on; if on it will be turned off. This toggling of statuses is the main action you can take in list menus.

Figure 4-5. Sample List Menu

```
Menu is CHSYDS            * ALIAS COMMAND SYSTEM *   Scenario is DEMO
------------------------------------------------------------------------
        CHOOSE THE SET OF VALID YARDS TO WHICH SHIPS MAY BE ASSIGNED
------------------------------------------------------------------------
    1.    AE IND                        13.    BELLING
    2.    AAA SF                        14.    BETH BA
    3.    AAA SO                        15.    BETH KEY
    4.    ADDSCO                        16.    BETH NJ
    5.    ALLIED                        17.    BETH SF
    6.    AMSHIP T                      18.    BETH SP
    7.    ARCWEL                        19.    BETHBEAU
    8.    ATKINSON                      20.    BETHBOST
    9.    ATLAN DD                      21.  * BIW
   10.    AVONDALE                      22.    BIW PORT
   11.    BAYSHIP                       23.    BOEINGFL
   12.    BELLHALT                      24.    BOEINGSE
------------------------------------------------------------------------

  COMMAND:
```

A list may have more candidates than will fit on a single display page (the sample one runs to four pages, only the first of which is shown). You may look at other pages by using the "+" and "-" commands to page up and down. List menus are the only ones in which paging is allowed.

List menus are also unusual in that you can effectively give more than one command on a single line. For example, if you wanted to change the statuses of candidates 4, 16, and 20 you could type "4,16,20" in response to the "COMMAND:" prompt instead of giving the number one at a time in response to three separate prompts. However, you can only change the statuses of candidates which appear on the displayed page.

The format of list menus is the standard one, with reminders and title at the top. Candidates appear in two numbered columns, with asterisks between the number and the candidate name indicating "on" status.

4.3.4 Help Menus

You can obtain on-line help from any of the three types of Command System menu by giving the "?" command. The response to this command will be a fourth variety of "menu", such as that shown in Figure 4-6, which presents all of the help options at your disposal. A menu is necessary since there are so many kinds of help available: descriptions of commands, of the subject the menu you gave the "?" command in is concerned with, help about its individual options, etc.

You cannot take any actions in help menus, you can only obtain help. To return to the menu you were in when you asked for help give the usual "^" command.

4.3.5 Summary

There are five basic things you can do in the Command System: run a processor, look at a different menu, set a parameter value, change the on/off status of members of a list, and

Figure 4-6. Help Menu

```
Standard commands usable in the help subsytem are:
 ^    pop back to previous menu        | /   pop to top-most menu
      refresh current screen           | *  display menu system map
 S    help for entire menu system      | ?   help for this help menu
 M    help for previous menu           | C   help for previous standard options
 #    help for line=# on previous menu |
```

HELP COMMAND: M

Give the command S in the help subsystem to find out about ALIAS and the
basic organization of its command system. You are now positioned in the
top, or highest level, ALIAS command menu. Other menus are spread down a
hierarchical tree which you may move through. All menus/functions may be
reached by a series of commands which you give starting from this menu. Current
functions include user environment set-up, an ability to call
non-ALIAS processors from within ALIAS, and a data base maintenance
subsystem.

HELP COMMAND: ^

get help. Processors can be run only from a choice menu. Choice menus also offer the main access to other menus, although list menus must be called up from their "owning" parameter menus. Help can be had anytime.

As a user of the Command System, you are likely to spend most of your time with choice menus; you will use the parameter and list menus only when you want to set up values for a scenario or make changes to your set-up. And since the choice menus are organized in a tree-like hierarchy, you are likely to use the "TOP" choice menu most often, as you travel from one branch to another.

4.4 AVAILABLE COMMANDS

As mentioned above, there are numeric, non-numeric, and setting types of commands. The meaning of the numeric commands changes with the menu; the form of settings are also determined by the individual menu. However, there are several "house-keeping" commands which perform additional useful functions. This section will present those commands.

4.4.1 Universal Commands

Universal commands are those which can be given (with the same meaning) in any menu. These include:

^ The upwards caret tells the Command System to "pop" or return to the menu it displayed just before the current one. Choice menu options are the principal means of moving down the Command System tree of menus; the caret is the principal means of moving back up again.

? The universal ALIAS help request. In the Command System the response will be a menu of the kinds of help which are available. The menu that appears is tailored to the type of menu the help request was issued in.

?x You can bypass the help menu, if you know the exact kind of help you want, by postscripting the "?" character with the request character which appears on the help menu for that kind of help. Thus ?* displays the system map, ?M tells you about the menu you're in, ?# tells you about option # on the menu you're in, etc. As you learn to use the system

make sure to ask for help from each different kind of menu so you become aware of the sorts of help that are available.

&   If for some reason the display has become garbled, or part of a menu has scrolled off the top, you can ask for a re-display of the menu by giving the "&" command.

/   Pops you to menu "TOP". Since the "TOP" menu occupies a special place at the top of the menu hierarchy, you are likely to want to get to it often. This special command jumps you back to it no matter how far down the tree the menu you are currently using is.

Q
E   Either of these commands can be used to terminate your Command System session. Not accepted in help menus.

{
}
{+
{-
{   The "bracket" commands ask for various services from the stored commands (or "command file") subsystem, which is discussed in more detail in Section 4.6. { invokes the stored command use facility; {name executes a particular name stored command; {+ starts the stored command creation process, while } ends it; and {- allows you to delete stored commands which you created. These commands are not accepted in help menus.

Note that only one command is accepted at a time, i.e. you are limited to one command per "COMMAND:" prompt, except in list menus.

4.4.2 Special List Menu Commands

In addition to those just presented, four additional commands are accepted in list menus:

+   This is a request for the next page of the list.

-     This is a request for the previous page of the list.

A   This sets the status of <u>all</u> candidates to "on", including those on pages of the list not currently shown. This command and the next one are exceptions to the general rule that you must be able to see a candidate to change its status.

N   This sets the status of all candidates to "off".

When making changes in a list menu, you are likely to want either to turn all candidates "on" or to have only a few "on". The "A "and "N" commands ease the tedium of performing this task.

For example, to convert a situation in which all candidates of a list are "on" to one in which only few are, the easiest means is by using "N" to turn all "off", followed by giving the numbers of those few which are to be "on".

4.4.3 Special Help Menu Commands

In addition to the universal commands, help menus respond to a set of commands which correspond to the kinds of help available:

M Requests display of the text description of the purpose of the choice, parameter, or list menu you're in.

* Requests that the system map (Figure 4-1 or similar to it) be displayed.

C Requests a description of each of the commands that are available in the choice, parameter, or list menu you're in.

S Requests that the ALIAS system description (an introductory summary of what ALIAS does and how to use it) be displayed.

? Given in this context, "?" causes a description of how the help subsystem works to be displayed.

where "#" is the number of one of the options on the choice, parameter, or list menu you're using. This causes a text description of the purpose and result of the option to be displayed.

4.5 ENVIRONMENT OPTIONS AND UTILITY PROGRAM ACCESS

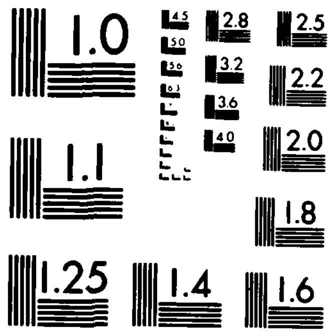
Two of the branches of the Command System menu tree do not lead to modules, but rather provide access to overall Command System control values and to utility programs.

Option 1 on menu "TOP" leads to the "ENVIRONMENT CONTROL" menu, whose first option in turn leads to the "USER ENVIRONMENT PARAMETERS" menu, shown in Figure 4-7. At present you may set three things in this menu: the type of terminal you are using, the printer you want your reports to come out on, and whether or not you want the reminder of commonly used commands to appear at the top of your Command System menus. Note that most modules

Figure 4-7. User Environment Parameters Menu

```
Menu is ENVRN              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
-------------------------------------------------------------------------
                        USER ENVIRONMENT PARAMETERS
 1.  TERMINAL TYPE            = SBRAIN        (HP2623,SBRAIN,HZ15,HZ14,HP2647,PC
 2.  DEVICE TO PRINT TO       = TERMINAL      (TERMINAL,DAISY,LP,PRINTER)
 3.  PRINT COMMANDS ON MENUS? = NO            (YES,NO)

  COMMAND:
```

AD-A150 424   ACQUISITION AND LOGISTICS INFORMATION AND ANALYSIS   3/4
SYSTEM (ALIAS) USER'S GUIDE(U) DECISION-SCIENCE
APPLICATIONS INC ARLINGTON VA M S CAREY ET AL.
UNCLASSIFIED   31 OCT 84 DSA-618 N00014-82-C-0813   F/G 15/5   NL


1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

consult the "DEVICE TO PRINT TO" parameter when they are ready to print, giving its setting a system-wide effect.

Option 2 on the "ENVIRONMENT CONTROL" choice menu is for system development and maintenance personnel, and should be used only by them.

Option 2 on menu "TOP" leads to the "HP PROGRAMS" choice menu (shown in Figure 4-8), in which you may request to use any of several programs generally available on the HP, such as the editors and graphics package. Note that only Data Base Administrators will be permitted to use RELATE (option 4) and MENU (option 5) this way.

It is particularly convenient to have access to an editor without leaving ALIAS, since this allows you to quickly perform such tasks as sending mail to other users and altering Force Level report format control files.

4.6 STORED COMMANDS

You may become impatient with a command system which prints out voluminous menus at each step. Stored commands offer a means of bypassing the menus.

A stored command is a series of ALIAS commands, perhaps given in a series of several menus, which can be invoked by name from a particular menu. Thus, if you want to use the TDP editor and you are in the "TOP" menu, you can give the command "{TDP" as an alternative to choosing option 2 and then option 2 in the HP PROGRAMS menu. The stored command gives the commands "2" and "/". It bypasses display of the menu, places you in TDP, and then returns you to menu "TOP" when you exit TDP.

Individual stored commands are not built-in parts of ALIAS, but rather are created by individual users as they see fit (all commands created by all users are available to everyone).

Figure 4-8. Utility Program Access Menu

```
Menu is HPROGS              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
------------------------------------------------------------------------
                              HP PROGRAMS
    1.  EDITOR
    2.  TDP
    3.  GRAPH
    4.  RELATE
    5.  MENU
    6.  SPOOK

  COMMAND:
```

If you know the name of the stored command you want to execute, you can just give it as a command prefixed by the "{" character. To find out what stored commands are available, give the "{" character all by itself.

Creation of a stored command is a straightforward task in which you step through the procedure you want stored. You must get to the menu you will want to use the command in after it is created, and then give the "{+" command to start the building process. You will be prompted for a name and a description, and then returned to the given choice, parameter, or list menu (a message will appear at the upper right corner to remind you that you are building a stored command).

Just step through the process you want to record. Note that stored commands apply only in the Command System, not in any other modules or processors. Thus, if you run the assigner as part of your procedure, any actions you take in the assigner will not be part of the stored command. Instead, when you later execute it, the command will "pause" as it executes the assigner, letting you perform any action you wish there, and resume when you leave the assigner.

Be aware that any changes made to parameter or list menu settings will become part of the stored command and thus will be repeated each time it is executed in the future.

When you have finished your procedure and are in the menu you want the stored command to leave you in when it completes, give the "}" command. This completes the creation process and makes the new command available.

You can invoke a stored command deletion processor by giving the "{-" command. You will only be allowed to delete

those that you created, not those created by others (unless you are a Data Base Administrator level user).

There are two important limitations of stored commands. First, they can only be executed from the Command System menu in which their creation began. Thus, one cannot execute the "TDP" command from the assigner's parameter menu. It is usually most convenient to start general-purpose stored commands from menu "TOP", since "TOP" can be jumped to immediately from anywhere using the "/" built-in command.

Second, any mistakes you make while constructing a stored command become a part of it, so be careful during the creation process!

4.7 SUMMARY: A GUIDED TOUR OF COMMAND SYSTEM MENUS

Table 4-1 summarizes the commands you can give while in the Command System. The remainder of this section presents the menus available in ALIAS Version 1.0, with running commentary on their purposes, in the same general format as used in Section 3.

Table 4-1. Summary of Command System Commands

COMMAND	USABLE	EFFECT
number	choice	executes choice menu option #
	list	toggles the status of list candidate #
#=value	param.	set parameter # to a new value
?	anywhere	display of a menu of help options
?*	"	display of the system map
?#	"	description of menu option #
?C	"	review of available commands
?M	"	description of the current menu
?S	"	description of ALIAS
??	"	description of how to use the help subsystem
Q or E	anywhere	exit the Command System; return to MPE
^	anywhere	return to the menu displayed before this one
	anywhere	re-display the current menu
/	anywhere	go back to menu TOP
{	anywhere	enter stored command execution subsystem
{name	anywhere	execute stored command "name" ("name" will be executable from only one menu)
{+	anywhere	start creating a new stored command
}	anywhere	complete creation of a stored command
{-	anywhere	delete stored commands
+	list menu	display the next page
-	list menu	display the previous page
A	list menu	turn "on" all list candidates
N	list menu	turn "off" all list candidates

On the facing page the TOP menu is shown in the top frame, while the system map pictorially displays the branches the TOP options lead to. Note that the map was displayed in response to the "?*" quick help request.

This tour will work its way down each branch of the menu tree in the order in which they appear on the top menu.

```
Menu is TOP                  * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu                 |  /   pop to top menu
 &   reprint with current values          | {+   build command file
 }   end command file building            | {    use command  file
 ?   use help processor                   |
_______________________________________________________________________
                          TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:?*
```

```
                          *** ALIAS MENU SYSTEM MAP ***
                                   ..............
c = choice menu                    : TOP CHOICE :
p = parameter menu                 :    MENU    :
l = list menu                      :............:
                                         |
     |-----------------|-----------------|-----------------|-----------------|
.......c.......   ......c.....   .......c......   .....c......   ........c.......
:    USER     :   : OTHER HP :   : SCENARIO   :   : ASSIGNER :   : FORCE LEVELS :
: ENVIRONMENT :   : PROGRAMS :   : CONTROL    :   :          :   :              :
:.............:   :..........:   :............:   :..........:   :..............:
       |                               ------------------|              |
.......p.......                        |                         .......p.......
: ENVIRONMENT :                  ......p.......                  : FORCE LEVEL :
: PARAMETERS  :                  :  ASSIGNER  :                  : PARAMETERS  :
:.............:                  : PARAMETERS :                  :.............:
                                 :............:                         |
                   |-----------------|-----------------|         .......l.......
                .....l.....   ......l.......   ......l.....      : REPAIR JOBS :
                : YARDS TO :  : CLASSES TO :   : JOB TYPES :     : CAUSING OUT-:
                :  ASSIGN  :  :   ASSIGN   :   : TO ASSIGN :     : OF FORCE    :
                :..........:  :............:   :...........:     :.............:

Hit RETURN for display of menu
```

The "Environment" branch, which starts with option 1 of TOP, is for setting some system-wide control variables which influence your work. You must go through the "Environment Control" menu to get to the "User Environment Parameters" menu which has these control values. Option 2 of the "Environment Control menu should be used only by software developers.

The "Environment Parameters" menu is the one to go to if ALIAS is not clearing your terminal's display before it types each menu; ALIAS may have made a poor guess about the make/model of the terminal you are using.

You can also choose which printer your hard copy output will appear on in this menu, and whether or not command "ticklers" will appear at the top of menu displays. They are appearing in this set of samples---note that option 3 started out set to "YES", was set to "NO" by a command, and then set back to "YES" again.

The "/" command causes a return to the TOP menu (so we can go down the next branch).

```
Menu is TOP                 * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu                | /    pop to top menu
 &    reprint with current values         | {+   build command file
 }    end command file building           | {    use command  file
 ?    use help processor                  |
--------------------------------------------------------------------------
                          TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:1

Menu is ENVIRC              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu                | /    pop to top menu
 &    reprint with current values         | {+   build command file
 }    end command file building           | {    use command  file
 ?    use help processor                  |
--------------------------------------------------------------------------
                              ENVIRONMENT CONTROL
    1.  ENVIRONMENT PARAMETERS
    2.  SET LPRNTS (DEBUG SWITCHES)

  COMMAND:1

Menu is ENVRN               * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu                | /    pop to top menu
 &    reprint with current values         | {+   build command file
 }    end command file building           | {    use command  file
 ?    use help processor                  |
--------------------------------------------------------------------------
                           USER ENVIRONMENT PARAMETERS
 1.  TERMINAL TYPE            = SBRAIN       (HP2623,SBRAIN,HZ15,HZ14,HP2647,PC
 2.  DEVICE TO PRINT TO       = TERMINAL     (TERMINAL,DAISY,LP,PRINTER)
 3.  PRINT COMMANDS ON MENUS? = YES          (YES,NO)

  COMMAND:?3

Allows you to turn off the tickler print at the top of each
menu of the most used ALIAS menu system commands.

  COMMAND:3=N
  COMMAND:3=Y
  COMMAND:/

Saving parameter settings...
```

The "Non-ALIAS Processors" branch provides a way to execute some commonly used programs without leaving the Command System. The "HP Programs" menu listing the ones available is displayed by choosing option 2 in TOP.

EDITOR and TDP are text editors, and GRAPH the HP business graphics package. RELATE and MENU run the DBMS, but security restrictions may prevent you from choosing these options. SPOOK is an HP program for managing spooled print files; it can be useful for checking the progress of any printouts you have started.

```
Menu is TOP               * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu               | /   pop to top menu
 &   reprint with current values        | {+  build command file
 }   end command file building          | {   use command  file
 ?   use help processor                 |
-----------------------------------------------------------------------
                        TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:2
```

```
Menu is HPROGS            * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu               | /   pop to top menu
 &   reprint with current values        | {+  build command file
 }   end command file building          | {   use command  file
 ?   use help processor                 |
-----------------------------------------------------------------------
                              HP PROGRAMS
    1.  EDITOR
    2.  TDP
    3.  GRAPH
    4.  RELATE
    5.  MENU
    6.  SPOOK

  COMMAND:/
```

The third branch leads directly to the Data Base Updating System. The DBU is the subject of Section 7.

The fourth branch services the assigner, and is the largest in terms of number of Command System menus (there are five). In the first of these you may choose either to run the assigner or to inspect its control variable values. The latter was chosen in the sample; the assigner's parameter menu is displayed at the bottom. The assigner looks at the values of the parameters displayed as it is running; you can customize its operations for your purposes by choosing the most appropriate set of values.

Three list menus are accessible from this parameter menu. The one with the list of "CANDIDATE SHIP YARDS" is requested in the sample.

```
Menu is TOP                 * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu             |  /   pop to top menu
 &    reprint with current values      |  {+  build command file
 }    end command file building        |  {   use command  file
 ?    use help processor               |
---------------------------------------------------------------------
                     TOP LEVEL ALIAS COMMAND MENU
   1.  CUSTOMIZE USER ENVIRONMENT
   2.  CALL NON-ALIAS PROCESSORS
   3.  DATA BASE UPDATING SYSTEM
   4.  MANUAL ASSIGNMENT EDITOR
   5.  FORCE LEVEL REPORT GENERATOR
   6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:4


Menu is ASSIGN              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu             |  /   pop to top menu
 &    reprint with current values      |  {+  build command file
 }    end command file building        |  {   use command  file
 ?    use help processor               |
---------------------------------------------------------------------
                    MANUAL ASSIGNER SPECIFICATIONS
   1.  ASSIGNER INITIALIZATION PARAMETERS
   2.  EXECUTE THE ASSIGNER

  COMMAND:1


Menu is ASNPRM              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu             |  /   pop to top menu
 &    reprint with current values      |  {+  build command file
 }    end command file building        |  {   use command  file
 ?    use help processor               |
---------------------------------------------------------------------
              MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                = FISCYR      (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE            = 10/ 1/1985  (MM/DD/YYYY)
 3.  ENDING DATE              =  9/31/1994  (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS     = LIST        (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES   = LIST        (ALL/LIST)
 6.  CANDIDATE JOB TYPES      = LIST        (ALL/LIST)
 7.  DISPLAY BASIS            = AWD         (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS             = START       (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE              = PROGRAM     (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION        = PROJ        (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER     = ALPHABETIC  (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER      = ALPHABETIC  (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH             = OFF         (ON,OFF)

  COMMAND:4=L
```

Only the first page of this (currently) four-page list is shown. The assigner will load and permit entry of assignments only for yards which are "turned on" in this list (only for BIW in this case).

Exiting this list brings the parameter menu back up again, where a request for the next list is issued.

```
Menu is CHSYDS              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu                  | /   pop to top menu
 &   print screen with current values      | A   all items are INCLUDED
 N   no items are INCLUDED                 | ?   help for this menu
 -   print previous list screen            | +   print followine list screen
-------------------------------------------------------------------------------
          CHOOSE THE SET OF VALID YARDS TO WHICH SHIPS MAY BE ASSIGNED
-------------------------------------------------------------------------------
      1.    A&E IND                         13.    BELLING
      2.    AAA SF                          14.    BETH BA
      3.    AAA SO                          15.    BETH KEY
      4.    ADDSCO                          16.    BETH NJ
      5.    ALLIED                          17.    BETH SF
      6.    AMSHIP T                        18.    BETH SP
      7.    ARCWEL                          19.    BETHBEAU
      8.    ATKINSON                        20.    BETHBOST
      9.    ATLAN DD                        21.  * BIW
     10.    AVONDALE                        22.    BIW PORT
     11.    BAYSHIP                         23.    BOEINGFL
     12.    BELLHALT                        24.    BOEINGSE
-------------------------------------------------------------------------------

  COMMAND:^

Saving list on/off settings...


Menu is ASNPRM              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu                  | /   pop to top menu
 &   reprint with current values           | {+  build command file
 }   end command file building             | {   use command  file
 ?   use help processor                    |
-------------------------------------------------------------------------------
               MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                 = FISCYR       (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE             = 10/ 1/1985   (MM/DD/YYYY)
 3.  ENDING DATE               =  9/31/1994   (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS      = LIST         (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES    = LIST         (ALL/LIST)
 6.  CANDIDATE JOB TYPES       = LIST         (ALL/LIST)
 7.  DISPLAY BASIS             = AWD          (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS              = START        (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE               = PROGRAM      (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION         = PROJ         (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER      = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER       = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH              = OFF          (ON,OFF)

  COMMAND:5=L
```

The "CANDIDATE SHIP CLASSES" list has the names of all ship classes for which there is data in the current scenario. As with the yards list, the assigner will load and permit entry of assignments only for classes which are "on". In this case, they all are.

```
Menu is CHSCLS              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
^    pop to previous menu              | /   pop to top menu
&    print screen with current values  | A   all items are INCLUDED
N    no items are INCLUDED             | ?   help for this menu
-    print previous list screen        | +   print followine list screen
--------------------------------------------------------------------------
        CHOOSE THE SET OF VALID SHIP CLASSES WHICH MAY BE ASSIGNED
--------------------------------------------------------------------------
    1. * AD-14                      13. * AGOS-1
    2. * AD-37                      14. * AK-279
    3. * AD-41                      15. * AK-280
    4. * AE                         16. * AK-286
    5. * AE-21                      17. * AK-85
    6. * AE-23                      18. * AO-105
    7. * AE-26                      19. * AO-143
    8. * AF-58                      20. * AO-177
    9. * AFDM                       21. * AO-177.2
   10. * AFS-1                      22. * AO-187
   11. * AFS-8                      23. * AO-22
   12. * AG                         24. * AO-51
--------------------------------------------------------------------------

  COMMAND:^

Saving list on/off settings...

Menu is ASNPRM              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
^    pop to previous menu              | /   pop to top menu
&    reprint with current values       | {+  build command file
}    end command file building         | {   use command  file
?    use help processor                |
--------------------------------------------------------------------------
          MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                 = FISCYR       (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE             = 10/ 1/1985   (MM/DD/YYYY)
 3.  ENDING DATE               =  9/31/1994   (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS      = LIST         (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES    = LIST         (ALL/LIST)
 6.  CANDIDATE JOB TYPES       = LIST         (ALL/LIST)
 7.  DISPLAY BASIS             = AWD          (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS              = START        (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE               = PROGRAM      (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION         = PROJ         (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER      = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER       = ALPHABETIC   (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH              = OFF          (ON,OFF)

  COMMAND:6=L
```

The "CANDIDATE JOB TYPES" list has the names of all the job types for which there is data in the current scenario.

The general capability to restrict the range of yards/ classes/job types the assigner will work with can be useful in two major ways. First, putting on restrictions which limit the assigner only to the things you are interested in reduces the volume of data you must page through, saving time. Second, it makes the assigner run faster because it need not load and update as much of the data base.

```
Menu is OBJTYP                * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu             | /   pop to top menu
 &    print screen with current values | A   all items are INCLUDED
 N    no items are INCLUDED            | ?   help for this menu
 -    print previous list screen       | +   print followine list screen
--------------------------------------------------------------------------
              CHOOSE THE SET OF VALID JOBS WHICH MAY BE ASSIGNED
--------------------------------------------------------------------------
    1. * CONV                           5. * REPAIR
    2. * NEWCON                         6. * SLEP
    3. * REACT                          7. * SLPCNV
    4. * REFUEL
--------------------------------------------------------------------------

  COMMAND:/

Saving list on/off settings...

Saving parameter settings...
```

The fifth Command System branch is for the Force Report Generators. Choosing option 5 in menu TOP causes the "FORCE LEVEL GENERATOR" menu to be displayed. This menus offers a choice between execution of the Force Level report generator, the Battle Group report generator, and inspection of the control variable values to which each of these programs will pay attention.

```
Menu is TOP                  * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu                 | /    pop to top menu
 &    reprint with current values          | {+   build command file
 }    end command file building            | {    use command  file
 ?    use help processor                   |
-----------------------------------------------------------------------
                           TOP LEVEL ALIAS COMMAND MENU
     1.   CUSTOMIZE USER ENVIRONMENT
     2.   CALL NON-ALIAS PROCESSORS
     3.   DATA BASE UPDATING SYSTEM
     4.   MANUAL ASSIGNMENT EDITOR
     5.   FORCE LEVEL REPORT GENERATOR
     6.   SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:5

Menu is FLRPTG               * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu                 | /    pop to top menu
 &    reprint with current values          | {+   build command file
 }    end command file building            | {    use command  file
 ?    use help processor                   |
-----------------------------------------------------------------------
                              FORCE LEVEL GENERATOR
     1.   FORCE LEVEL REPORT INITIALIZATION PARAMETERS
     2.   EXECUTE FORCE REPORT GENERATOR
     3.   PREPARE BATTLE GROUP REPORT

  COMMAND:1
```

These report generators have a somewhat shorter list of control variables, and only a single list menu. The list menu displays the job type codes for repair jobs. An asterisk next to a code (implying status = "on") indicates that the given repair job will cause ships undergoing this repair to be undeployable while the repair work is being performed.

```
Menu is FLREPT              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu              |  /   pop to top menu
 &    reprint with current values       |  {+  build command file
 }    end command file building         |  {   use command  file
 ?    use help processor                |
------------------------------------------------------------------------
           FORCE LEVEL AND BATTLEGROUP REPORT GENERATOR PARAMETERS
 1.   KEEP REPORT ON-LINE       = YES         (YES,NO)
 2.   REPORT START DATE         =  1/ 1/1986  (MM/DD/YYYY)
 3.   REPORT END DATE           = 12/31/1999  (MM/DD/YYYY)
 4.   RETIRE SHIPS BY           = LIFE        (LIFE,DATE)
 5.   TIME PERIOD LENGTH        = CALYR       (DAY,WEEK,MONTH,QTR,CALYR)
 6.   IN FORCE DAY              = END         (BEGIN,END)
 7.   PROGRAM MILESTONE         = APPROP      (APPROP,AWD,DELIV)
 8.   OUT OF FORCE REPAIR JOBS  = LIST        (ALL/LIST)

  COMMAND:8=L

Menu is FLREPT              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu              |  /   pop to top menu
 &    print screen with current values  |  A   all items are INCLUDED
 N    no items are INCLUDED             |  ?   help for this menu
 -    print previous list screen        |  +   print followine list screen
------------------------------------------------------------------------
          REPAIR JOBS THAT REMOVE A SHIP FROM FORCE DURING EXECUTION
------------------------------------------------------------------------
    1.    REFUEL                        3. * SLEP
    2.    REPAIR
------------------------------------------------------------------------

  COMMAND:/

Saving list on/off settings...

Saving parameter settings...
```

The last branch of Command System menus, accessed through option 6 of menu TOP, services the scenario system. During a session you may finish using a scenario and wish to use another. You need not leave the Command System and re-run it to do this---just choose option 1 on the Scenario System menu. Option 2 lets you create new scenarios, while 3 lets you delete them; with 6 you can change their composition (perhaps drawing some data into the one you are working with from another). Options 4 and 5 lists the scenarios in existence either to the terminal or the printer.

```
Menu is TOP                  * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu                 | /    pop to top menu
 &   reprint with current values          | {+   build command file
 }   end command file building            | {    use command  file
 ?   use help processor                   |
```

TOP LEVEL ALIAS COMMAND MENU

1. CUSTOMIZE USER ENVIRONMENT
2. CALL NON-ALIAS PROCESSORS
3. DATA BASE UPDATING SYSTEM
4. MANUAL ASSIGNMENT EDITOR
5. FORCE LEVEL REPORT GENERATOR
6. SCENARIO CHOICE/MAKEUP SYSTEM

COMMAND:6

```
Menu is SCEN                 * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^   pop to previous menu                 | /    pop to top menu
 &   reprint with current values          | {+   build command file
 }   end command file building            | {    use command  file
 ?   use help processor                   |
```

SCENARIO SYSTEM MENU

1. CHOOSE A DIFFERENT SCENARIO TO WORK WITH
2. CREATE A NEW SCENARIO
3. DELETE CURRENTLY EXISTING SCENARIOS
4. LIST CURRENTLY EXISTING SCENARIOS
5. SEND LIST OF EXISTING SCENARIOS TO LINE PRINTER
6. MODIFY THE MAKEUP OF AN EXISTING SCENARIO

COMMAND:/

This tour of Command System menus has included all offered by Version 1.0 with the exception of the help menus. To find out more about the assigner, force level, or scenario system menus, turn to Sections 6, 8 and 9.

```
Menu is TOP                * ALIAS COMMAND SYSTEM *   Scenario is DEMO
 ^    pop to previous menu               |  /    pop to top menu
 &    reprint with current values        |  {+   build command file
 }    end command file building          |  {    use command  file
 ?    use help processor                 |
------------------------------------------------------------------------
                      TOP LEVEL ALIAS COMMAND MENU
    1.  CUSTOMIZE USER ENVIRONMENT
    2.  CALL NON-ALIAS PROCESSORS
    3.  DATA BASE UPDATING SYSTEM
    4.  MANUAL ASSIGNMENT EDITOR
    5.  FORCE LEVEL REPORT GENERATOR
    6.  SCENARIO CHOICE/MAKEUP SYSTEM

  COMMAND:Q

Sure you want to terminate your ALIAS session?Y

END OF PROGRAM
```

5.0 DATA FLOW AND THE DATA BASE

ALIAS is a "data driven" software system, meaning that much of its capability comes from the way in which its data is stored and manipulated, rather than from the internal logic of its computer programs. This makes it flexible, since data is easy to change as new needs arise, while programs are hard to change.

Consider the construction schedules for the ships of a POM projection. Often these will be generated by a run of the assigner, using standard time intervals between milestone dates for each type of ship. Do the schedules just have the year of award and these planned intervals? No, they have explicit milestone dates, as computed by the assigner from the planning factors (intervals). This makes them usable by processors other than the assigner, especially since they are identical in format to the schedules for jobs currently underway and for historical jobs. And if you do not like the milestones the assigner generates for any given ship, you are free to change them and make your changes stick.

Data useful for more than one purpose is always stored in the data base, and in low-level, detailed formats which allow you to fine-tune. This standardization ensures that any changes you make in the underlying data will be reflected in all the reports you produce, not just one.

Figure 5-1 diagrams the data flows involving the data base. Most of the incoming volume is data input resulting from reports from the field, while most of the output is in the form of reports. Of crucial importance to you as a user, however, are the (usually) low-volume changes made to data base values during the course of an analysis. The ability to make these changes to the picture of the ship acquisition world which the data base holds makes it possible for you to implement a wide variety of assumptions about the course of future events. In order to use this capability, you must understand how the ALIAS data base is

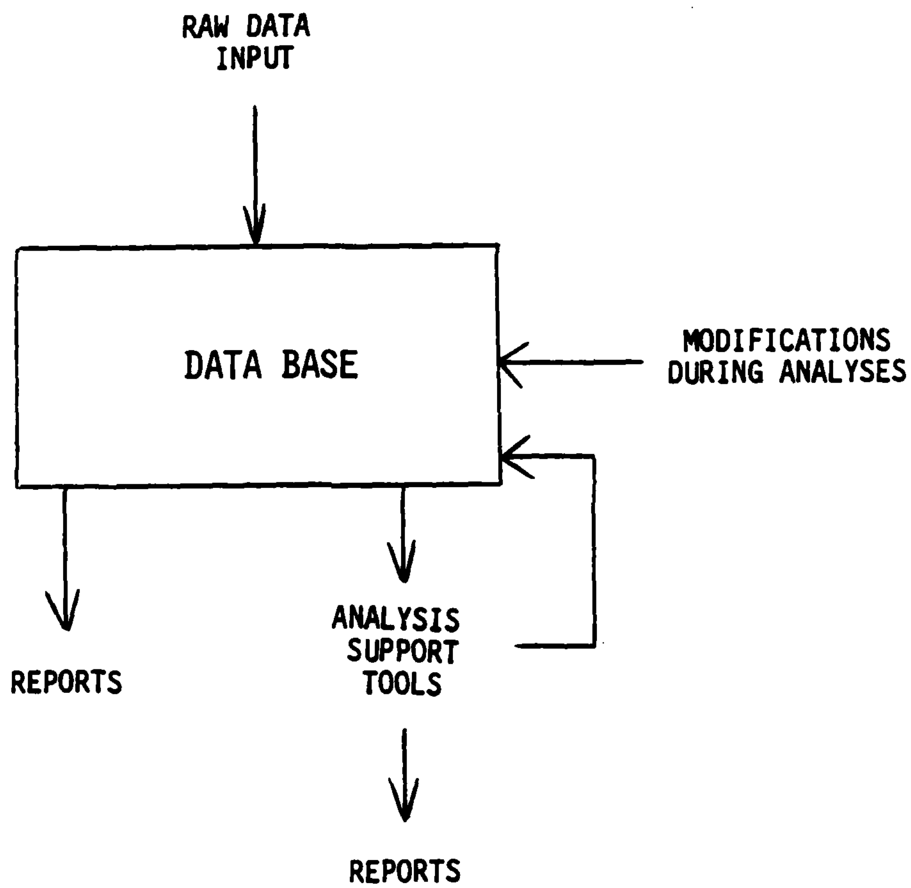


Figure 5-1. Data Flow Summary

organized. This knowledge will also help you construct customized reports.

5.1 DATA BASE ORGANIZATION

ALIAS has a central data base and a number of peripheral data bases, as diagrammed in Figure 5-2. The central data base contains all the data which you can update using the Data Base Updating system, and is the only one managed by the scenario system. Peripheral data bases are those created by individuals or teams to support their own projects; there may be plans to include these in the central data base at some point, or they may be for temporary use. Examples of peripheral data bases are NAVSHIPSO's material and equipment production capacity and lead time data bases (large and heavily used), and the per-unit ship cost estimate data base used in the sample session in Section 3 (small and temporary).

Because all ALIAS data bases are organized along relational lines, it is often possible to combine information in central and peripheral data bases to produce reports (this was done to produce the cost report in the sample session). The distinction between central and peripheral data bases focuses on how data gets into them---data input to the central DB must meet high standards of internal consistency, while owners of peripheral DBs set their own standards. The distinction is less important for report production because the RELATE commands used to produce reports are capable of combining data from a wide variety of sources.

Lack of consistency can cause some problems for report production, however. For example, if one data base uses the code "DD-963" and another "DD 963" for the same class of ship, RELATE will be unable to combine their data.

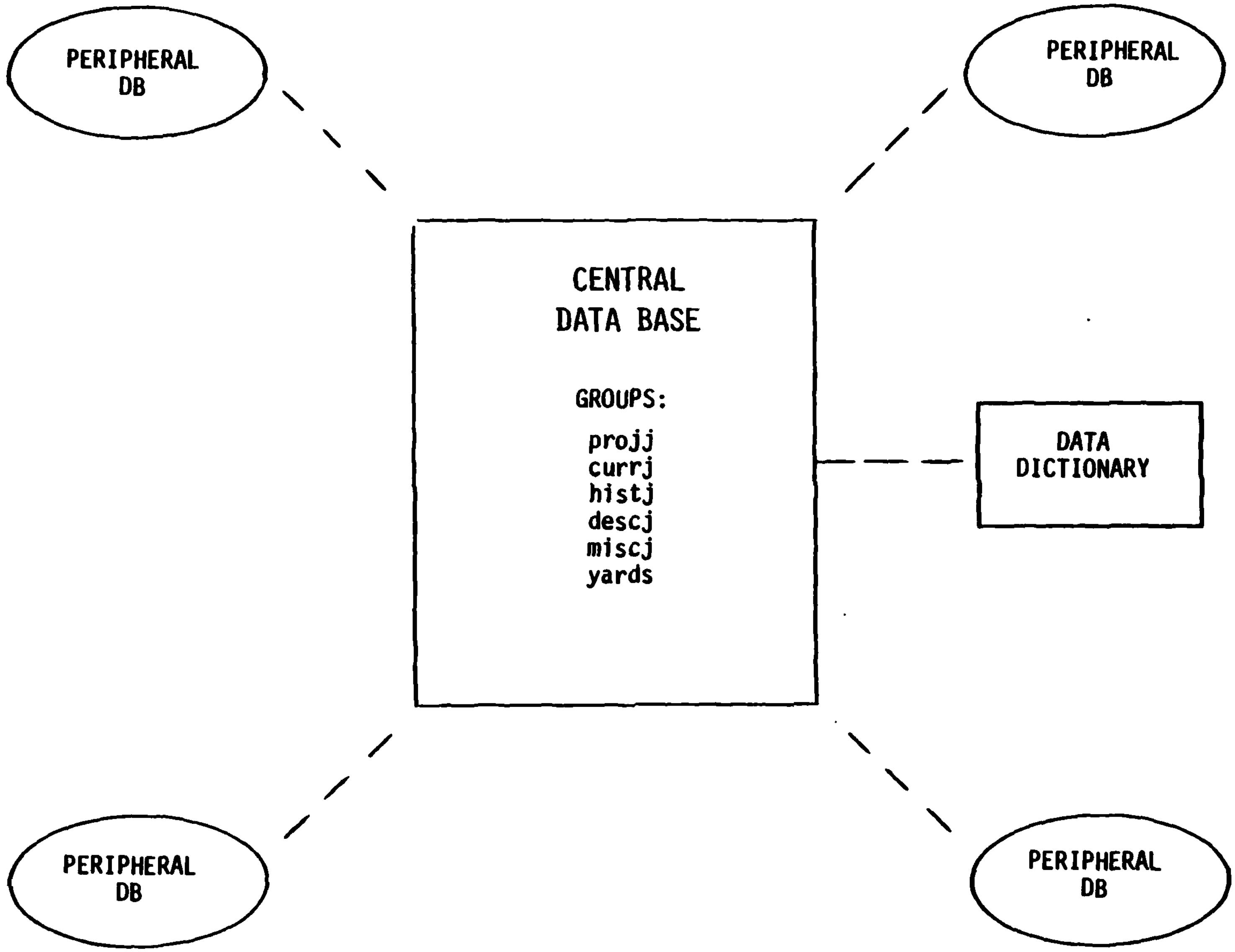


Figure 5-2. Data Base Organization

5.1.1 Data Storage Conventions

All ALIAS data bases are composed of relations, which are RELATE data files. Each relation requires two HP 3000 files, one for the actual data and one for indexing (sort ordering) information. The first is named after the relation (up to 7 characters, e.g., SHDESC), while the second has a "Q" appended to its name (e.g., SHDESCQ). To access a relation using RELATE you need only know its primary name.

Each relation is a table composed of rows and columns. The columns are named and are often called "fields", while the rows are called "records" or "tuples". Generally each field is meant to describe some aspect of something; thus a row is a description of one instance of that thing. For example, the SHDESC relation holds (partial) physical descriptions of ship classes. The fields are things like CLASS and LAUN_TONS; a record holds the description for a single class.

There is a unique key field set for each relation. The set of values found in these fields in any given record will be unique. The nature of the unique key field set for a given relation influences how queries of it should be constructed. The unique key fields for the SHDESC relation are SCENARIO,CLASS, DATADATE,ENTRY_DATE.

When DATADATE appears in a key field set it means that more than one instance of the "substantive" subset of the key field values can be in the relation simultaneously. In the SHDESC example there can be more than one record for any given class (for a given scenario). In such cases you must structure your queries to get the record with the latest data (if that is what you desire). The only central data relation which does NOT have DATADATE in its unique key field set is ncjodat.projj.

Relations which hold data about a given subject tend to be stored together in separate HP file groups. For example, data

having to do with individual ships currently under construction is kept in relations in the .currj group. If you want to know about schedules for such ships, you should refer to relation ncjodat.currj.

To fully utilize the resources of the data base, you must know what relations it contains and what fields each relation has. This will be the subject of Section 5.3.

5.2 DATA BASE ACCESS METHODS

There are four major ways of extracting data from the data base. These are queries, report generation, using the Data Base Updating System, and writing programs. There is only one way of changing the contents of the data base: the Data Base Updating System (see Section 7).

ALIAS security restricts the ways in which you can use these methods: queries and report generation must be done interactively while running RELATE, and you may only run RELATE when logged on under your "R" user name (e.g., JOHNR, not JOHNA). You will not be allowed to change any data base values when operating in this mode.

The DBU may only be run from the ALIAS Command System, which in turn can be run only when you are logged on under your "A" user name. Custom programs which you write must likewise be run in your "A" persona.

5.2.1 Queries

A query can be thought of as a question you ask the data base using interactive RELATE commands. It can range from "Tell me the names of all the ship classes the Navy has built" to "Tell me how many ships are projected for SCN award next year and how much each is projected to cost" to very complicated questions. To make a query you must decide what information is required,

identify fields in particular relations which hold this information, and then combine the relations' contents in a way that will give you the desired printed output on your terminal.

The output is generally in relational form, i.e. in a table of rows and columns, but you can also obtain summary statistics about any given table. To find out the delivery dates of new ships projected for delivery in the first half of fiscal 1990 you would give the following commands when running RELATE:

```
  1)OPEN FILE NCJODAT.PROJJ;MODE=SHARED
  2)SELECT SCENARIO,CLASS,HULL,DELIVERY BY DELIVERY &
 .1&)WHERE DELIVERY>="10/1/1989" AND DELIVERY<="3/30/1990"&
 .2&)AND SCENARIO="DEMO"
WARNING: TEMPORARY INDEX WILL BE CREATED TO SATISFY BY CLAUSE.
  3)PRINT
```

and would get this as a result:

DELIVERY	SCENARIO	CLASS	HULL
10/30/1989	DEMO	SSN-688	756
10/31/1989	DEMO	CG-47	61
10/31/1989	DEMO	MSH-1	7
11/01/1989	DEMO	T-AO-187	194
11/01/1989	DEMO	AO-187	191
11/01/1989	DEMO	AFDM	1
11/02/1989	DEMO	MCM-1	7
12/15/1989	DEMO	AE	3
12/21/1989	DEMO	T-ACS	8
12/30/1989	DEMO	CG-47	62
12/31/1989	DEMO	SSN-688	757
1/02/1990	DEMO	MCM-1	8
1/30/1990	DEMO	MSH-1	8
3/01/1990	DEMO	DD-963	999

14 LINES PRINTED.

That was an example of a query from a single table. A similar one requiring data from two tables might request that the standard service life of each ship be given as well as its delivery date:

```
 4)OPEN FILE SHLIFE.MISCJ;MODE=SHARED
 5)SET PATH NCJODAT
 6)SELECT SCENARIO,CLASS,HULL,DELIVERY,SHLIFE.LIFE, &
.1&)SHLIFE.TIMUNT BY DELIVERY &
.2&)WHERE DELIVERY>="10/1/1989" AND DELIVERY<="3/30/1990"&
.3&) AND SCENARIO="DEMO" AND SHLIFE.SCENARIO=NCJODAT.SCENARIO &
.4&) AND SHLIFE.CLASS=NCJODAT.CLASS
WARNING: TEMPORARY INDEX WILL BE CREATED TO SATISFY BY CLAUSE.
 7)PRINT
```

DELIVERY	SCENARIO	CLASS	HULL	LIFE	TIMUNT
10/30/1989	DEMO	SSN-688	756	30	CALYR
10/31/1989	DEMO	CG-47	61	30	CALYR
10/31/1989	DEMO	MSH-1	7	30	CALYR
11/01/1989	DEMO	T-AO-187	194	30	CALYR
11/01/1989	DEMO	AO-187	191	30	CALYR
11/01/1989	DEMO	AO-187	191	30	CALYR
11/01/1989	DEMO	AFDM	1	30	CALYR
11/02/1989	DEMO	MCM-1	7	30	CALYR
12/15/1989	DEMO	AE	3	30	CALYR
12/21/1989	DEMO	T-ACS	8	30	CALYR
12/30/1989	DEMO	CG-47	62	30	CALYR
12/31/1989	DEMO	SSN-688	757	30	CALYR
1/02/1990	DEMO	MCM-1	8	30	CALYR
1/30/1990	DEMO	MSH-1	8	30	CALYR
3/01/1990	DEMO	DD-963	999	30	CALYR

15 LINES PRINTED.

The additional conditions "AND SHLIFE.SCENARIO=SCENARIO AND SHLIFE.CLASS=CLASS" tell RELATE that for every record it finds meeting the delivery date and scenario conditions in the ncjodat. projj relation, it should find a record in the shlife.miscj relation with matching SCENARIO and CLASS field values. RELATE effectively glues the contents of the two tables together side-by-side (all classes were arbitrarily assigned the same service lives in this sample data).

Notice that this second query resulted in a 15-line table instead of the 14 lines from the first query; AO-187:191 was printed twice. This happened because shlife.miscj is one of the relations which has DATADATE included in its unique key field

set. There were two records in shlife.miscj for the AO-187 class, with two different DATADATE values. RELATE produced a result including the data from each such "duplicate" record. To avoid this sort of problem, the following alternative type of query should be given:

```
  8)SELECT SCENARIO,CLASS,HULL,DELIVERY,SHLIFE.LIFE, &
 .1&)SHLIFE.TIMUNT BY DELIVERY &
 .2&)WHERE DELIVERY>="10/1/1989" AND DELIVERY<="3/30/1990"&
 .3&) AND SCENARIO="DEMO" AND SHLIFE.SCENARIO=NCJODAT.SCENARIO &
 .4&) AND SHLIFE.CLASS=NCJODAT.CLASS &
 .5&) AND SHLIFE.DATADATE=$MAX(SHLIFE.DATADATE BY &
 .6&) SHLIFE.SCENARIO,SHLIFE.CLASS)
WARNING: TEMPORARY INDEX WILL BE CREATED TO SATISFY BY CLAUSE.
  9)PRINT
```

DELIVERY	SCENARIO	CLASS	HULL	LIFE	TIMUNT
10/30/1989	DEMO	SSN-688	756	30	CALYR
10/31/1989	DEMO	CG-47	61	30	CALYR
10/31/1989	DEMO	MSH-1	7	30	CALYR
11/01/1989	DEMO	T-AO-187	194	30	CALYR
11/01/1989	DEMO	AO-187	191	30	CALYR
11/01/1989	DEMO	AFDM	1	30	CALYR
11/02/1989	DEMO	MCM-1	7	30	CALYR
12/15/1989	DEMO	AE	3	30	CALYR
12/21/1989	DEMO	T-ACS	8	30	CALYR
12/30/1989	DEMO	CG-47	62	30	CALYR
12/31/1989	DEMO	SSN-688	757	30	CALYR
1/02/1990	DEMO	MCM-1	8	30	CALYR
1/30/1990	DEMO	MSH-1	8	30	CALYR
3/01/1990	DEMO	DD-963	999	30	CALYR

14 LINES PRINTED.

The extra condition SHLIFE.DATADATE=$MAX(SHLIFE.DATADATE BY SHLIFE.SCENARIO,SHLIFE.CLASS) tells RELATE to use only those shlife.miscj records which have the largest (latest) DATADATE value for each different set of values for SCENARIO,CLASS. The result is then as expected. This condition should be used in queries of any relation which includes DATADATE in its unique key field set and for which you want only the latest data.

The query results so far have been in the usual relational form; they might have been obtained just by printing the contents of an existing relation, had one existed which held the exact data desired. Suppose that only the number of ships to be delivered in the first half of 1990 was desired, not data about each one. In the examples above it is easily seen that the number is "14", but if you were making a query where the expected answer was more like "2000" you might not want to watch 2000 lines of output scroll across your terminal just to find out how many there were. To get the count only, give a query similar to the following one:

```
 8)SELECT NUM=$COUNT(HULL WHERE SCENARIO="DEMO" AND &
.1&)DELIVERY>="10/1/1989" AND DELIVERY<="3/30/1990")
 9)PRINT

  NUM

   14

1 LINE PRINTED.
```

Notice that the conditions on the count must go inside the parentheses of the $COUNT aggregate.

To learn more about how to construct queries using RELATE, see especially the sections on the SELECT and PRINT commands in the RELATE Reference Manual.

5.2.2 Reports

The sample queries of the last section produced output that was fairly rough-and-ready. Although you have some control over the order and placement of the columns of output produced by the PRINT statement through the use of "switches" (see the RELATE manual), more sophisticated formatting requires the use of the CREATE report generator. CREATE is part of the RELATE software

package and is accessed from RELATE by giving the REPORT command. CREATE does not enhance your ability to make queries at all---you must set up the query using OPEN FILE and/or SELECT statements prior to giving the REPORT command---but does greatly increase your control over the appearance of the output.

Suppose you wanted to know delivery dates for cruisers and guided missile destroyers being delivered in the first half of fiscal 1993, and that you wanted the dates grouped by ship class in a simple but presentable fashion. The following commands might be used to do your bidding:

```
  1)OPEN FILE NCJODAT.PROJJ;MODE=SHARED
  2)SELECT SCENARIO,CLASS,HULL,DELIVERY BY CLASS,DELIVERY &
 .1&) WHERE SCENARIO="DEMO" AND DELIVERY>="10/1/1992" AND &
 .2&) DELIVERY<="3/30/1993"
WARNING: TEMPORARY INDEX WILL BE CREATED TO SATISFY BY CLAUSE.
  3)REPORT
  1>PAGE HEADING=("Scenario ";TAB=40),(SCENARIO;TAB=49),&
 .1&>(DATE;NEWLINE=1;TAB=40),&
 .2&>("AAW DELIVERIES BY CLASS, FIRST HALF 1993";NEWLINE=2;
 .3&>  JUSTIFY=CENTER)
  2>FIELDS=(SCENARIO;NOPRINT),(CLASS;NOPRINT;BREAK=10),&
 .1&>(HULL;TAB=10;NOHEADING),(DELIVERY;TAB=20;NOHEADING)
  3>GROUP HEADING 10=(NEWLINE=3),(CLASS;TAB=18),&
 .1&> ("DELIVERIES";TAB=+1),(NEWLINE=2),("HULL";TAB=17;UNDERL),&
 .2&> ("DATE";TAB=27;UNDERLINE)
  4>GO
```

with the result:

```
                                        Scenario DEMO
                                        12/12/84

     AAW DELIVERIES BY CLASS, FIRST HALF 1993

                  CG DELIVERIES

                 HULL      DATE
                 ----      ----
                   60      01/16/1993
                   61      03/28/1993
```

DDG DELIVERIES

HULL	DATE
57	12/20/1992
58	2/28/1993
59	3/15/1993

Note particularly that output can be grouped according to the changing values of one or more fields by using the BREAK- option in the FIELD command in combination with the GROUP HEADING and GROUP FOOTING commands. In the example a new group was produced every time CREATE came to a new class name; the name of the class was placed in the group heading instead of in a column.

Also note that CREATE commands can be placed in an editor-type file along with OPEN FILE and SELECT statements. In this form the report the statements produce can be run over and over again from RELATE using the EXECUTE command. The work of setting up and entering report commands thus needs to be done only once.

See the CREATE manual for more information about how to generate formatted reports.

5.2.3 Queries Via the Data Base Updating System

A limited amount of querying can be done using the Search command of the Data Base Updating system (DBU). You can type in field values which target records must match, and the DBU will attempt to find such records for you by constructing a custom SELECT statement. You can then print the result using the P command.

However, you are generally limited to retrieval of data from a single relation, cannot put on restrictions such as the range of valid dates specified in the above queries, and can only look at the output one record at a time on-screen. This method is thus most useful when your target is one or a few records whose values you want to change rather than just look at.

See Section 7 for more information about the DBU.

5.2.4 Writing Programs to Make Queries

If you are a programmer, you can construct even more sophisticated queries than RELATE is capable of by writing programs which access the data base through RELATE's Host Language Interface. Before using one of the more traditional languages such as FORTRAN or COBOL, see if your task can be done using the BUILDER screen language interpreter which was used to implement the DBU---if so, the time you will need to spend to implement your program will be reduced by a large factor.

See the ALIAS Maintenance Guide for more information about creating programs in the ALIAS environment.

5.3 STRUCTURE OF THE DATA BASE

This section will describe the relations and fields of the central ALIAS data base in some detail. It is meant as a reference for your use in constructing queries. Most of the information presented here can also be found in the ALIAS data dictionary, which always contains the most up-to-date description of the data base structure.

5.3.1 The Data Dictionary

The ALIAS data dictionary is a data base in its own right. It consists of several relations stored in the .db group (e.g., filinfo.db). The data dictionary serves three purposes:

1) It is a reference which can be used to discover how the data base is set up.

2) The DBU consults it for rules which your data base updates must conform to.

3) It is used as the basis for creation of the central data base relations---a special program creates the relations from their data dictionary structural descriptions, ensuring that the data dictionary and the actual data base agree with one another.

Table 5-1 lists the relations in the data dictionary and describes their purpose. Table 5-2 is an annotated list of the fields in selected dictionary relations.

5.3.2 The Data Base

Table 5-3 is an annotated listing of the groups that centrtions in these groups are necessarily in the central data base). Table 5-4 is an annotated listing of relations in each group. Table 5-5 lists the fields in each relation, while Table 5-6 describes the purpose of each field. Tables 5-5 and 5-6 may be reproduced from the contents of the fldfile.db and fldesc.db relations whenever you desire.

Note that any field which appears with the same name in multiple relations will always have the same data format, enabling you to reliably join data from several relations.

Table 5-1. Annotated List of Data Dictionary Relations

RELATION	PURPOSE
DBFLDS	A list of all the fields in the central data base, including information such as data type (integer, alpha, etc.) and size. Though a field may appear in several relations, it will always conform to the standards set here.
FILDESC	Textual descriptions of the purpose of each data relation.
FILINDX	Contains the permanent indexes which exist for each data relation. Used by the makfile.dba relation creation processor to create the indexes at file creation time.
FILINFO	A list of the relations in the central data base, along with brief descriptions of purpose and parameters used by various ALIAS processors.
FILJOIN	Data base integrity maintenance rules; used by the DBU to validate updates.
FILPRIV	Data base security privilege information. Lists relations and which users may perform which DBU functions on them. These security specifications apply only to DBU operations.
FILSCRN	A cross reference describing which DBU screens serve each data relation.
FLDESC	Textual descriptions of the purpose of each data base field.
FLDFILE	A cross reference describing which fields are in each data relation. Used by the makfile.dba relation creation processor to determine file structure at creation time.
FLDUSER	A list of the ALIAS processors which use each field in each file. Useful for determination of the magnitude of impact of changes to a given field in a given relation.
REPTEX	A list of available report generators.
SCRFLDS	A list of the fields appearing on each DBU screen, including cross references between screen variable names and associated relation field names.
SCRPRIV	DBU security privilege information. Lists DBU screens and which users may perform which functions when using them.

Table 5-1. Annotated List of Data Dictionary Relations

RELATION	PURPOSE
SREPTS	A companion to REPTEX, this lists the report generators which are available from each DBU data screen.

Table 5-2. Annotated List of Data Dictionary Fields

RELATION	FIELD	PURPOSE
DBFLDS	FIELDNAME	Name of the field the record describes.
	DATATYPE	Data type of the field.
	MAXVALUE	Gives the maximum size of the data value the field can hold. The biggest number for numeric fields, number of characters for alphanumeric.
	FORMATSPEC	The special conditions section of the field creation syntax used during creation of files the field is in.
	UNITYPE	Units the field is expressed in. This is ar important reference for understanding of the contents of some numeric fields.
	PRINTLEN	Field width specification for file creation statements. Must agree with MAXVALUE for alphanumeric fields.
	LEGLFILE	Blank for fields whose acceptable values are NOT specified by a list in the legal values reference library (set of relations in the .legals group). For legals-type fields, name of the .legals relation holding the appropriate list.
FILINDX	FILE	Name of the relation the index is on.
	GROUP	Group the relation resides in.
	INUM	Number the index has (determines the order in which indexes are created and thus which index is picked by, e.g., SET INDEX 1.
	INDEX	Specification of the fields in the index in CREATE INDEX syntax, including the UNARY specification if any.
FILINFO	FILE	Name of the relation the record describes.
	GROUP	Group the relation resides in.
	FAMILY	Expanded phrase describing the data group the relation belongs to.
	DOMAIN	PUBLIC or PRIVATE.
	CODERESP	NAVSEA/NAVSHIPSO code responsible for data update.
	RELNUM	ID number of the relation. An obsolete field no longer used.
	SCENSET	Name of the data group the relation belongs to for scenario purposes. Generally the name of the HP file group the relation is in, but "SYSTEM" for relations which are not scenario dependent (e.g. data dictionary relations or .legals relations), and "PARAMS" for relations forming the special command system data base created by program mnug.

Table 5-2. Annotated List of Data Dictionary Fields

RELATION	FIELD	PURPOSE
	SCENUM	Scenario system ID number of the relation. Determines the storage column for the relation's scenario field values in the table which cross references overall scenario name to key values (table stored in relsnl.sysrw, readable only by ALIAS). MUST NOT BE CHANGED ONCE DETERMINED FOR A GIVEN RELATION. Must be unique for each relation.
	SCRSET	Name of the data group the relation belongs to for DBU purposes. Must be "LEGALS" for .legals relations; the DBU uses the value of this field to determine which relations to open as the legal values reference library.
	SCRNUM	DBU ID number of the relation. Determines the storage location in the DBU file management system's extra data segment for the relation's actual partition. Must be unique.
	SCRPROC	ID of the RELATE son process that the DBU will create the relation's actual partition in. Relations may be redistributed among more processes if overflows occur as ALIAS is expanded.
	SCRINDX	Number of the index which should be SET to when the relation is opened by the DBU.
	QDESCRIP	A one-line description of the purpose of the relation.
FILJOIN	SRCFILE SRCGROUP SRCFIELD	Name of a relation and field within the relation which is to be half of one clause in a join condition which must be satisfied before an addition or modification to the SRC relation can be made.
	TGTFILE TGTGROUP TGTFIELD	Name of a relation and field within it which is the other half of this join condition clause. The clause will take the form AND SRCFILE.SRCFIELD=TGTFILE.TGTFIELD, and there will be as many clauses in the condition as there are records in filjoin.db with identical sets of values for srcfile,srcgroup,tgtfile, and tgtgroup.
	FLEV	Numeric field whose value determines the order in which clauses appear in the condition.
	JOINTYPE	One of REQUIRED (data update can't be made if a SELECT made with the condition as its WHERE clause finds no records), RECOMMENDED (a warning is issued by the DBU if the SELECT finds no records), or IN_ONE:A, IN_ONE:B, etc. If the condition fails for the two relations forming IN_ONE:A, then the

Table 5-2. Annotated List of Data Dictionary Fields

RELATION	FIELD	PURPOSE
		DBU moves on to try IN_ONE:B, and so on.
	JOINMSG	Text placed in DBU's warning or failure message regarding update request.
	DATANAME	Text placed in DBU's verification message during subsidiary data deletion; should describle the kind of data in the SRC relation.
FILPRIV	FILE	Name of the relation the security specification applies to.
	GROUP	Name of the group the relation is stored in.
	USERNAME	Name of the user the specification applies to. Limit of one specification per relation-user combination.
	READ	"Y" if user can read the relation's contents.
	ADD	"Y" if user can add records to the relation.
	DELETE	"Y" if user can delete records.
	UPDATE	"Y" if user can update records.
	MODIFY	"Y" if user can modify records.
FILSCRN	FILE	Name of the relation the screen serves.
	GROUP	Group the relation is stored in.
	SCREEN	Name of the DBU screen used to update the relation.
FLDFILE	FILE	Name of the relation the field appears in.
	GROUP	Name of the group the relation is stored in.
	FIELDNAME	Name of a field found in the relation.
	FLDNUM	Column number in which the field appears. SCENARIO must always have a FLDNUM value of 1.
	CODERESP	NAVSEA/NAVSHIPSO code responsible for update of the field's data.
	PRIMSOURCE	Primary source of the field's data.
	SECSOURCE	Secondary source for the field's data.
FLDUSER	FILE	Name of the relation the field is in.
	GROUP	Group the relation is stored in.
	FIELDNAME	Name of a field in the relation.
	USERNAME	Name of the ALIAS module or report generator which makes use of and/or changes values of the field.
	USERTYPE	Nature of the user---program, screen, or report generator.
SCRFLDS	SCREEN	Name of the DBU screen the field appears in.
	SCRVAR	Name of the screen variable serving the field.

Table 5-2. Annotated List of Data Dictionary Fields

RELATION	FIELD	PURPOSE
	FIELDNAME	Name of the field in its relation.
	FILE	Name of the relation.
	GROUP	Name of the group the relation is in.
	FLDTYPE	Role the field plays in the relation and on the screen---key field (i.e. part of the relation's unique key field set), ordinary data field, or legals field (one whose value must be on an approved list stored in .legals).
SCRPRIV	SCREEN	Name of the DBU screen the privileges are for.
	USERNAME	Name of the user the privileges are for.
	SEEIT	"Y" if user is allowed to use the given screen.

Table 5-3. Annotated List of Central Data Base Groups

GROUP	PURPOSE
CURRJ	Current SCN and repair job data, where "current" means those appropriated but not yet delivered.
DB	Data dictionary relations.
DESCJ	Projected job description data (planning factors).
HISTJ	Historical SCN and repair job data, where "historical" means those already delivered.
LEGALS	All relations comprising the legal values reference library.
MISCJ	Miscellaneous information, primarily ship-class level data (physical descriptions, service lives).
MNUREL MAKMENU	Repository for the relations which hold Command System parameters and lists. These data are scenario-dependent and managed by the DBU.
PROJJ	Projected SCN and repair job data, i.e. for those jobs not yet appropriated.
YARDS	Shipyard data.

Table 5-5. List of Field Appearing in Each Relation

NCJODAT	HISTJ	SCENARIO
NCJOCOM	HISTJ	SCENARIO
NCJOCOM	CURRJ	SCENARIO
NCJODAT	PROJJ	SCENARIO
NCJOCOM	PROJJ	SCENARIO
STRT103	DB	DUMMY
DEACT	MISCJ	SCENARIO
DEACOM	MISCJ	SCENARIO
SHCOMT	MISCJ	SCENARIO
SHLIFE	MISCJ	SCENARIO
SHDESC	MISCJ	SCENARIO
YARDID	YARDS	SCENARIO
REJOCOM	CURRJ	SCENARIO
REJODAT	PROJJ	SCENARIO
REJOCOM	PROJJ	SCENARIO
REJOCOM	HISTJ	SCENARIO
NCJDAT	DESCJ	SCENARIO
YARDCOM	YARDS	SCENARIO
NCJDCOM	DESCJ	SCENARIO
NCJODAT	CURRJ	SCENARIO
REJODAT	HISTJ	SCENARIO
REJODAT	CURRJ	SCENARIO
NCJODAT	HISTJ	CLASS
NCJOCOM	HISTJ	CLASS
NCJOCOM	CURRJ	CLASS
NCJODAT	PROJJ	CLASS
NCJOCOM	PROJJ	CLASS
DEACT	MISCJ	CLASS
DEACOM	MISCJ	CLASS
SHCOMT	MISCJ	CLASS
SHLIFE	MISCJ	CLASS
SHDESC	MISCJ	CLASS
YARDID	YARDS	YARD
REJOCOM	CURRJ	CLASS
REJODAT	PROJJ	CLASS
REJOCOM	PROJJ	CLASS
REJOCOM	HISTJ	CLASS
NCJDAT	DESCJ	CLASS
YARDCOM	YARDS	YARD
NCJDCOM	DESCJ	CLASS
NCJODAT	CURRJ	CLASS
REJODAT	HISTJ	CLASS
REJODAT	CURRJ	CLASS
NCJODAT	HISTJ	HULL
NCJOCOM	HISTJ	HULL
NCJOCOM	CURRJ	HULL
NCJODAT	PROJJ	HULL
NCJOCOM	PROJJ	HULL

Table 5-5. List of Field Appearing in Each Relation

DEACT	MISCJ	HULL
DEACOM	MISCJ	HULL
SHCOMT	MISCJ	HULL
SHLIFE	MISCJ	LIFE
SHDESC	MISCJ	PRNTNAME
YARDID	YARDS	YARDNAME
REJOCOM	CURRJ	HULL
REJODAT	PROJJ	HULL
REJOCOM	PROJJ	HULL
REJOCOM	HISTJ	HULL
NCJDAT	DESCJ	NCJOBT
YARDCOM	YARDS	DATADATE
NCJDCOM	DESCJ	NCJOBT
NCJODAT	CURRJ	HULL
REJODAT	HISTJ	HULL
REJODAT	CURRJ	HULL
NCJODAT	HISTJ	COMNUM
NCJOCOM	HISTJ	COMNUM
NCJOCOM	CURRJ	COMNUM
NCJODAT	PROJJ	COMNUM
NCJOCOM	PROJJ	COMNUM
DEACT	MISCJ	COMNUM
DEACOM	MISCJ	COMNUM
SHCOMT	MISCJ	DATADATE
SHLIFE	MISCJ	TIMUNT
SHDESC	MISCJ	HULL
YARDID	YARDS	PARENT
REJOCOM	CURRJ	JOBID
REJODAT	PROJJ	JOBID
REJOCOM	PROJJ	JOBID
REJOCOM	HISTJ	JOBID
NCJDAT	DESCJ	YARD
YARDCOM	YARDS	ENTRYDATE
NCJDCOM	DESCJ	YARD
NCJODAT	CURRJ	COMNUM
REJODAT	HISTJ	JOBID
REJODAT	CURRJ	JOBID
NCJOCOM	HISTJ	DATADATE
NCJOCOM	CURRJ	DATADATE
NCJOCOM	PROJJ	DATADATE
DEACOM	MISCJ	DATADATE
DEACT	MISCJ	DISPOSN
SHCOMT	MISCJ	ENTRYDATE
SHLIFE	MISCJ	DATADATE
SHDESC	MISCJ	CUSTOMER
YARDID	YARDS	ADDRESS
REJOCOM	CURRJ	DATADATE
REJOCOM	PROJJ	DATADATE
REJOCOM	HISTJ	DATADATE
NCJDAT	DESCJ	JSTYP

Table 5-5. List of Field Appearing in Each Relation

NCJODAT	HISTJ	YARD
NCJODAT	PROJJ	YARD
REJODAT	PROJJ	YARD
YARDCOM	YARDS	LNUM
NCJDCOM	DESCJ	JSTYP
NCJODAT	CURRJ	YARD
REJODAT	HISTJ	REJOBT
REJODAT	CURRJ	REJOBT
NCJODAT	HISTJ	NCJOBT
NCJOCOM	HISTJ	ENTRYDATE
NCJOCOM	CURRJ	ENTRYDATE
NCJODAT	PROJJ	NCJOBT
NCJOCOM	PROJJ	ENTRYDATE
DEACOM	MISCJ	ENTRYDATE
DEACT	MISCJ	DEACT
SHCOMT	MISCJ	LNUM
SHLIFE	MISCJ	ENTRYDATE
SHDESC	MISCJ	LENGTHWL
YARDID	YARDS	STATE
REJOCOM	CURRJ	ENTRYDATE
REJODAT	PROJJ	REJOBT
REJOCOM	PROJJ	ENTRYDATE
REJOCOM	HISTJ	ENTRYDATE
NCJDAT	DESCJ	COMNUM
YARDCOM	YARDS	COMMENT
NCJDCOM	DESCJ	DATADATE
NCJODAT	CURRJ	NCJOBT
REJODAT	HISTJ	YARD
REJODAT	CURRJ	YARD
NCJODAT	HISTJ	JSTYP
NCJOCOM	HISTJ	LNUM
NCJOCOM	CURRJ	LNUM
NCJODAT	PROJJ	JSTYP
NCJOCOM	PROJJ	LNUM
DEACT	MISCJ	DATADATE
DEACOM	MISCJ	LNUM
SHCOMT	MISCJ	COMMENT
SHLIFE	MISCJ	ENTRYBY
SHDESC	MISCJ	LENGTHOA
YARDID	YARDS	FSCM
REJOCOM	CURRJ	LNUM
REJODAT	PROJJ	JSTYP
REJOCOM	PROJJ	LNUM
REJOCOM	HISTJ	LNUM
NCJDAT	DESCJ	CMETHD
YARDCOM	YARDS	ENTRYBY
NCJDCOM	DESCJ	ENTRYDATE
NCJODAT	CURRJ	JSTYP
REJODAT	HISTJ	JSTYP
REJODAT	CURRJ	JSTYP

Table 5-5. List of Field Appearing in Each Relation

NCJODAT	HISTJ	CUSTOMER
NCJOCOM	HISTJ	COMMENT
NCJOCOM	CURRJ	COMMENT
NCJODAT	PROJJ	CUSTOMER
NCJOCOM	PROJJ	COMMENT
DEACT	MISCJ	DATETYPE
DEACOM	MISCJ	COMMENT
SHCOMT	MISCJ	ENTRYBY
SHDESC	MISCJ	BEAMWL
YARDID	YARDS	SMSA
REJOCOM	CURRJ	COMMENT
REJODAT	PROJJ	CUSTOMER
REJOCOM	PROJJ	COMMENT
REJOCOM	HISTJ	COMMENT
NCJDAT	DESCJ	CUSTOMER
NCJDCOM	DESCJ	ENTRYBY
NCJODAT	CURRJ	CUSTOMER
REJODAT	HISTJ	CUSTOMER
REJODAT	CURRJ	CUSTOMER
NCJODAT	HISTJ	SHIPNAME
NCJOCOM	HISTJ	ENTRYBY
NCJOCOM	CURRJ	ENTRYBY
NCJODAT	PROJJ	SHIPNAME
NCJOCOM	PROJJ	ENTRYBY
DEACT	MISCJ	DATASOURCE
DEACOM	MISCJ	ENTRYBY
SHDESC	MISCJ	BEAMOA
YARDID	YARDS	CODE071
REJOCOM	CURRJ	ENTRYBY
REJODAT	PROJJ	APPROP
REJOCOM	PROJJ	ENTRYBY
REJOCOM	HISTJ	ENTRYBY
NCJDCOM	DESCJ	COMMENT
NCJODAT	CURRJ	SHIPNAME
REJODAT	HISTJ	APPROP
REJODAT	CURRJ	APPROP
NCJDAT	DESCJ	COMPLEXGRP
NCJODAT	HISTJ	CMETHD
NCJODAT	PROJJ	CMETHD
DEACT	MISCJ	ENTRYBY
SHDESC	MISCJ	LAUNHITE
YARDID	YARDS	SUPSHIP
REJODAT	PROJJ	AWARD
NCJDAT	DESCJ	DEFLTAWDAY
NCJOCOM	HISTJ	DATETYPE
NCJDCOM	DESCJ	LNUM
NCJOCOM	CURRJ	DATETYPE
NCJODAT	CURRJ	CMETHD
REJOCOM	HISTJ	DATETYPE
REJOCOM	CURRJ	DATETYPE

Table 5-5. List of Field Appearing in Each Relation

REJODAT	HISTJ	AWARD
REJODAT	CURRJ	AWARD
NCJODAT	HISTJ	APPROP
NCJODAT	PROJJ	APPROP
DEACT	MISCJ	ENTRYDATE
SHDESC	MISCJ	LITEHITE
YARDID	YARDS	HOMEPORT
REJODAT	PROJJ	ARRIV
NCJDAT	DESCJ	DAYSADDED
NCJODAT	CURRJ	APPROP
REJODAT	HISTJ	ARRIV
REJODAT	CURRJ	ARRIV
NCJODAT	HISTJ	AWARD
NCJODAT	PROJJ	AWARD
SHDESC	MISCJ	FULLHITE
YARDID	YARDS	WATERWAY
REJODAT	PROJJ	START
NCJDAT	DESCJ	APPRCPAWD
NCJODAT	CURRJ	AWARD
REJODAT	HISTJ	START
REJODAT	CURRJ	START
NCJODAT	HISTJ	START
NCJODAT	PROJJ	START
SHDESC	MISCJ	LAUNDRAFT
YARDID	YARDS	OCEAN
REJODAT	PROJJ	DRYDOCK
NCJDAT	DESCJ	AWDST
NCJODAT	CURRJ	START
REJODAT	HISTJ	DRYDOCK
REJODAT	CURRJ	DRYDOCK
NCJODAT	HISTJ	KEEL
NCJODAT	PROJJ	KEEL
SHDESC	MISCJ	LITEDRAFT
YARDID	YARDS	TOTACRES
REJODAT	PROJJ	LAUNCH
NCJDAT	DESCJ	STKL
NCJODAT	CURRJ	KEEL
REJODAT	HISTJ	LAUNCH
REJODAT	CURRJ	LAUNCH
NCJODAT	HISTJ	LAUNCH
NCJODAT	PROJJ	LAUNCH
SHDESC	MISCJ	FULLDRAFT
YARDID	YARDS	DEVACRES
REJODAT	PROJJ	DELIVERY
NCJDAT	DESCJ	KLLN
NCJODAT	CURRJ	LAUNCH
REJODAT	HISTJ	DELIVERY
REJODAT	CURRJ	DELIVERY
NCJODAT	HISTJ	DELIVERY
NCJODAT	PROJJ	DELIVERY

Table 5-5. List of Field Appearing in Each Relation

SHDESC	MISCJ	LAUNTONS
YARDID	YARDS	USEDACRES
REJODAT	PROJJ	DAYSADDED
NCJDAT	DESCJ	LNDL
NCJODAT	CURRJ	DELIVERY
REJODAT	HISTJ	DAYSADDED
REJODAT	CURRJ	DAYSADDED
NCJODAT	HISTJ	COMMISSION
NCJODAT	PROJJ	COMMISSION
SHDESC	MISCJ	LITETONS
YARDID	YARDS	CHWIDTH
REJODAT	PROJJ	ASNORDER
NCJDAT	DESCJ	DLCOM
NCJODAT	CURRJ	COMMISSION
REJODAT	HISTJ	ASNORDER
REJODAT	CURRJ	ASNORDER
NCJODAT	HISTJ	DAYSADDED
NCJODAT	PROJJ	DAYSADDED
SHDESC	MISCJ	FULLTONS
YARDID	YARDS	CHLENGTH
REJODAT	PROJJ	DATADATE
NCJDAT	DESCJ	TIMUNT
NCJODAT	CURRJ	DAYSADDED
REJODAT	HISTJ	DATADATE
REJODAT	CURRJ	DATADATE
NCJODAT	HISTJ	ASNORDER
NCJODAT	PROJJ	ASNORDER
SHDESC	MISCJ	DATASOURCE
YARDID	YARDS	CHHEIGHT
REJODAT	PROJJ	DATASOURCE
NCJDAT	DESCJ	DATASOURCE
NCJODAT	CURRJ	ASNORDER
REJODAT	HISTJ	DATETYPE
REJODAT	CURRJ	DATETYPE
NCJODAT	HISTJ	DATADATE
NCJODAT	PROJJ	DATADATE
SHDESC	MISCJ	DATADATE
YARDID	YARDS	CHDEPTH
REJODAT	PROJJ	ENTRYBY
NCJDAT	DESCJ	DATADATE
NCJODAT	CURRJ	DATADATE
REJODAT	HISTJ	DATASOURCE
REJODAT	CURRJ	DATASOURCE
NCJODAT	HISTJ	DATETYPE
NCJODAT	PROJJ	DATASOURCE
SHDESC	MISCJ	ENTRYDATE
YARDID	YARDS	RAILDIST
REJODAT	PROJJ	ENTRYDATE
NCJDAT	DESCJ	ENTRYDATE
NCJODAT	CURRJ	DATETYPE

Table 5-5. List of Field Appearing in Each Relation

REJODAT	HISTJ	ENTRYBY
REJODAT	CURRJ	ENTRYBY
NCJODAT	HISTJ	DATASOURCE
NCJODAT	PROJJ	ENTRYBY
SHDESC	MISCJ	ENTRYBY
YARDID	YARDS	LOCKNAME
NCJDAT	DESCJ	ENTRYBY
NCJODAT	CURRJ	DATASOURCE
REJODAT	HISTJ	ENTRYDATE
REJODAT	CURRJ	ENTRYDATE
NCJODAT	HISTJ	ENTRYBY
NCJODAT	PROJJ	ENTRYDATE
YARDID	YARDS	MINLOCKLEN
NCJODAT	CURRJ	ENTRYBY
NCJODAT	HISTJ	ENTRYDATE
NCJODAT	PROJJ	AUTOMOD
YARDID	YARDS	MINLOCKWID
NCJODAT	CURRJ	ENTRYDATE
NCJODAT	PROJJ	PROGVAR1
YARDID	YARDS	OPENED
NCJODAT	PROJJ	PROGVAR2
YARDID	YARDS	CLOSED
NCJODAT	PROJJ	SUBRELUMAP
YARDID	YARDS	BECAME
YARDID	YARDS	DATASOURCE
YARDID	YARDS	DATADATE
YARDID	YARDS	ENTRYDATE
YARDID	YARDS	ENTRYBY

Table 5-6. Annotated List of Central Data Base Fields

FIELDNAME	DESCRIPTION
ADDRESS	Street address of the establishment, including state and zip.
APPROP	Job appropriation date.
APPROP_AWD	Length of time required between appropriation and award.
ARRIV	Date repair job arrives in yard.
ASNORDER	Special variable used only by the assigner module.
AUTOMOD	If yes, assigner may automatically modify schedule dates.
AWARD	Job award date.
AWD_ST	Length of time required between job award and start dates.
BEAM_OA	Overall beam of ship, including flight deck.
BEAM_WL	Beam of ship at the waterline.
BECAME	Name of yard that yard became after closing, if any.
CHDEPTH	Depth of channel at shallowest point in sea access lane.
CHHEIGHT	Maximum height of a ship above waterline to traverse sea access 1
CHLENGTH	Maximum length a ship may be and still traverse sea access lane.
CHWIDTH	Maximum beam a ship may have (waterline) and still traverse
CLASS	Ship class name, e.g. DDG-51
CLOSED	Date establishment closed its doors, under given name.
CMETHD	Construction/work method used on ship job/.

Table 5-6. Annotated List of Central Data Base Fields

FIELDNAME	DESCRIPTION
CODE071	Yard identification code number used by SEA 071.
COMMENT	A 65-character line of comments.
COMMENT	Holds text information, usually comments about other data.
COMMISSION	Date ship is commissioned or recommissioned.
COMNUM	Ship's commissioning number. =1 except for reactivation jobs.
COMPLEXGRP	Complexity group ships of class belong to for assigner update
COMPLEXGRP	purposes
CUSTOMER	Code name indicating the buyer of the given ship job.
DATADATE	Date the data in the data record was obtained.
DATASOURCE	Source data in record was gathered from.
DATETYPE	Code indicating whether dates are actuals or projections.
DAYSADDED	Number of days added to ship's life by jhob. 0 for newcon jobs.
DEACT	Ship's deactivation date for given commissioning.
DEFLTAWDAY	Default award day. Used by the assigner to determine the
DELIVERY	Date ship is delivered.
DEVACRES	Number of developed acres owned by yard at given site.
DISPOSN	Disposition of ship upon deactivation, e.g. SCRAP.
DL_COM	Amount of time required between ship delivery and commissioning.
DRYDOCK	Date ship is drydocked.
DUMMY	Dummy field; not used for any purpose.

Table 5-6. Annotated List of Central Data Base Fields

FIELDNAME	DESCRIPTION
ENTRY_BY	Name of user who originally entered or updated data in this recrd
ENTRY_DATE	Date this data was entered or updated (not modified).
FSCM	Federal code number identifying the company/plant.
FULLNAME	Full descriptive name for legal code value.
FULL_DRAFT	Draft of ship when fully loaded.
FULL_HITE	Height of ship above waterline when fully loaded.
FULL_TONS	Displacement of ship when fully loaded.
HOMEPORT	Home port code name.
HULL	Individual ship's hull number.
JOBID	A code number uniquely identifying the given job, in combination
JOBTYP	Ship job type code, e.g. NEWCON or REACT.
JSTYP	Job sequence type code, e.g. LEAD or ORDFOL.
KEEL	Date keel is laid, or for reactivation/conversion, of drydocking.
KL_LN	Time required between laying of keel and launch.
LAUNCH	Date ship is launched or undocked.
LAUN_DRAFT	Draft of ship when it is launched.
LAUN_HITE	Height of ship above waterline when it is launched.
LAUN_TONS	Displacement of ship when it is launched.
LENGTH_OA	Overall length of ship, including flight deck.
LENGTH_WL	Length of ship at the waterline.
LIFE	Standard service life of ships of the class, in the absence
LITE_DRAFT	Draft of the ship when light.

Table 5-6. Annotated List of Central Data Base Fields

FIELDNAME	DESCRIPTION
LITE_HITE	Height of the ship above waterline when light.
LITE_TONS	Displacement of the ship when light.
LNUM	Line number counter for comments or other text.
LN_DL	Amount of time required between ship launch and delivery.
LOCKNAME	Name of minimum-size lock in yard's sea access channel.
MINLOCKLEN	Length of the minimum-size lock in the yard's sea access channel.
MINLOCKWID	Width of the minimum-size lock in the yard's sea access channel.
NCJOBT	Job type code for new construction jobs.
OCEAN	Legal-value name of ocean the yard's access channel leads to.
OPENED	Date the facility opened.
PARENT	Name of the establishment's parent corporation, if any.
PRNTNAME	Name to print on reports
RAILDIST	Distance to nearest rail link.
REJOBT	Job type code for repair-type jobs.
SCENARIO	Name of ALIAS scenario given data record belongs to.
SHIPNAME	Full name of ship.
SMSA	Standard Metropolitan Statistical Area code number.
START	Date of start of work on job.
STATE	Abbreviated name for state the establishment is in.
ST_KL	Amount of time required between start of work and keel-laying.

Table 5-6. Annotated List of Central Data Base Fields

FIELDNAME	DESCRIPTION
SUPSHIP	Code name for SUPSHIP office supervising the yard.
TIMUNT	Indicates the units that an associated time-oriented data
TIMUNT	field is expressed in; e.g. MONTHS
TOTACRES	Total acres owned by the establishment.
USEDACRES	Number of acres now in use by the establishment.
WATERWAY	Name of the waterway that is the yard's principal sea access
YARD	Shipyard name, short (common usage) version.
YARDNAME	Full name of the shipyard.
YESNO	Legal-value library field name which must be YES or NO.

6.0 USING SCENARIOS

Each relation in the central ALIAS data base is effectively partitioned into sections called "scenarios". The partitioning is done according to the values of the SCENARIO field, the first one in each relation. All records belonging to a given scenario will have that scenario's name in their SCENARIO field.

This arrangement was made to allow data for multiple studies to coexist in the single ALIAS data base simultaneously. Since alteration of data values is a key tool used in ALIAS analyses, it would have been crippling to force everyone to use the same data.

One problem with this arrangement has to do with data base updating. Say there are 10 scenarios in existence. Each of these will have data for current jobs in the relations in the .CURRJ group. Say that the scenarios differ only in their projected jobs data, and that their owners all want to use the latest current job data. Does current job data updating have to be done 10 times, once per scenario?

The answer depends on the structure of the scenarios. There must be data for each scenario in each relation, but there is a distinction to be made between data being used by a scenario and data belonging to a scenario. When a data record belongs to a scenario it has that scenario's name in its scenario field. But other scenarios can use this data record, the only restrictions being that they cannot change the record, and they cannot have any records of their own in the same relation.

Each scenario has a map which tells it which data records to use in each central data base relation. The map consists of a list of SCENARIO field values, one per relation. If the map has a given scenario's own name for a given relation, then the data with that name in the SCENARIO field in that relation belongs to that scenario, and the scenario is said to be a "direct" user of

that data. If the map has some other scenario's name, then the scenario is an "indirect" user with read privileges for the data but no change privileges.

For the 10-scenario example above, the current job updating would have to be done only once (say for scenario "MAIN") as long as the other 9 scenarios were indirect users of MAIN's current job data.

The "structure" of a scenario is just its pattern of direct/indirect use over data base relations.

You must know how to create, use, and modify scenarios in order to make effective use of the ALIAS Command System and modules, and you must understand how a scenario's structure influences the manner in which you make data base queries. These are the subjects of this section.

6.1 CONTROLLING SCENARIOS

Access to the capabilities which let you use and manipulate scenarios is centered in the Command System scenario menu, reproduced in Figure 6-1. There are five basic functions: choice, creation, deletion, modification, and listing.

6.1.1 Picking a Scenario to Use

Whenever you use the Command System you must have a "current" scenario; all the data you see will be associated with it. You will be prompted for the name of the scenario you want to start out with when you first run the Command System, but you can use a different one at any time by choosing option 1 in the Command System's scenario menu. This will result in the menu shown in Figure 6-2, in which you are expected to give the name of the scenario you wish to change to. If you have access to it and no one else is using it at the time the change will be made and you will be returned to the Command System.

Figure 6-1. Command System Scenario Choice Menu

```
Menu is SCEN              * ALIAS COMMAND SYSTEM *   Scenario is DEMO
----------------------------------------------------------------------
                         SCENARIO SYSTEM MENU
    1.  CHOOSE A DIFFERENT SCENARIO TO WORK WITH
    2.  CREATE A NEW SCENARIO
    3.  DELETE CURRENTLY EXISTING SCENARIOS
    4.  LIST CURRENTLY EXISTING SCENARIOS
    5.  SEND LIST OF EXISTING SCENARIOS TO LINE PRINTER
    6.  MODIFY THE MAKEUP OF AN EXISTING SCENARIO

  COMMAND:

======================================================================
```

Figure 6-2. Menu for Choice of Scenario to Use

```
                                              Current scenario is
Scenario choice options include:
 ?      provides help                  |  ^    exits scenario choice system
 @      lists existing scenarios       |  +    moves you into scenario
 @name  lists the composition          |         creation menu
        of the 'name' scenario         |        re-displays this menu
 name   makes the 'name' scenario      |
        your current scenario          |
----------------------------------------------------------------------
                        SCENARIO CHOICE MENU
Name of scenario to use, or Command character:
```

You can also give any of the commands listed at the top of Figure 6-2 in response to the command prompt. These will be covered in detail in Section 6.1.6.

6.1.2 Creating a New Scenario

You will probably want to create new scenarios whenever you start a new study, or when you want to construct a variant of an existing study. Doing this allows you to make data changes freely while retaining the data from older studies in the exact form you left it in.

Before sitting down to perform the creation, give some thought to the structure you want the new scenario to have. Most scenarios start out as a combination of copies of some of the data in one or more existing scenarios and patterns of indirect-access use for the rest of their data. You will almost never want to create a scenario that is a completely blank slate, since you will have an enormous data entry task to get it ready for use.

First decide which scenario(s) you want to take data from. It is best if it all comes from a single existing scenario, since you can then be sure that the data you start out with is internally consistent. You are free to combine data from several scenarios, but you may have to correct consistency problems before you will be able to use the new scenario. For example, if you take yard data from scenario A, and projected jobs data from scenario B, and B contains a job to be done at yard X for which B has data but A does not, you will either have to delete the job or add data for yard X to your new scenario. A report of such problems will be produced at the close of the creation process to help you identify and correct them.

Make your data source decisions on the basis of the data groups listed in Table 5-3 (or Table 6-1 below). During creation everything is done in terms of sets of relations, which basically

Table 6-1. List of Data Base Sets and Relations

SET	RELATION
CURRJ	NCJOCOM.CURRJ
	NCJODAT.CURRJ
	REJOCOM.CURRJ
	REJODAT.CURRJ
DESCJ	NCJDAT.DESCJ
	NCJDCOM.DESCJ
HISTJ	NCJOCOM.HISTJ
	NCJODAT.HISTJ
	REJOCOM.HISTJ
	REJODAT.HISTJ
MISCJ	DEACOM.MISCJ
	DEACT.MISCJ
	SHCOMT.MISCJ
	SHDESC.MISCJ
	SHLIFE.MISCJ
MNUREL	ASNPRM.MNUREL
	ENVRN.MNUREL
	FLREPT.MNUREL
	LCCREF.MNUREL
	VALCLS.MNUREL
	VALYDS.MNUREL
	VLJTYP.MNUREL
	VLRJOB.MNUREL
PROJJ	NCJOCOM.PROJJ
	NCJODAT.PROJJ
	REJOCOM.PROJJ
	REJODAT.PROJJ
YARDS	YARDCOM.YARDS
	YARDID.YARDS

correspond to the contents of these groups, with all the relations in the same set getting the same treatment. If you need to treat relations within the same set differently, you can use the scenario modification machinery to do it after the creation is complete.

Once you have decided where the data in each set will come from, decide (by set) whether you will need to make changes to it. For sets where no changes are needed, you can use the data in the source scenario indirectly instead of having a copy of it made. This makes the creation process go faster, economizes on disk space, and may reduce your data entry burden. But if you know that the owner of the source scenario will be making changes to the data which you will not want to show up in your new scenario, or if you anticipate needing to make changes yourself, a copy of the source data must be made that will belong to your scenario and that you will use directly.

All of these decisions are reversible using the scenario modification machinery; in cases where you are uncertain of the best strategy, keep in mind that you can change your mind later.

Once you have made your decision, choose option 2 on the Command System's scenario menu. This will bring up the menu shown in Figure 6-3, in which you will be prompted for the name of your new scenario (it must be unique and less than 12 characters), for a description of it (up to 24 lines), and for some decisions about the access others will have to it. You will be asked if users other than yourself should be allowed to look at the scenario; if you say yes, any ALIAS user will be able to "run" the scenario and inspect its data. You cannot really prevent inspection of the data by valid ALIAS users in any case, since queries can be made using RELATE, but you can deny them the conveniences of the Command System and DBU by saying no.

Figure 6-3. Scenario Creation Menu, Stage 1

```
                                     Current scenario is
Scenario creation options include:
 ?      provides help                 | ^     exits scenario choice system
 @      lists existing scenarios      |      re-displays this menu
 name   specifies the name of your    | @name lists the composition
         new scenario                 |        of the 'name' scenario
-------------------------------------------------------------------------
                     SCENARIO CREATION--STAGE 1

NAME of new scenario, or COMMAND: DEMO

Do you wish to allow all ALIAS users to
examine the contents of this scenario? Y

Do you wish to allow all ALIAS users to
modify the contents of this scenario? Y

Please give a description of this scenario.  Terminate it with a '//' line.
.......................................................................
scenario which goes with user manual sample session
//
```

You will also be asked if other users should be allowed to change the scenario's contents. If you say yes, then others will not only be able to see your data, but will be able to change it as well. If you say no, ALIAS security is sufficient to prevent anyone else from making changes to your data.

After you respond to these questions the menu shown in Figure 6-4 will appear (this one is a copy from the sample session in Section 3). Here you will be prompted for your scenario structure decisions. You will be asked for the source scenario for data for each set. If you want to start with a clean slate in any set, respond with the name of your new scenario. You will be asked if you will want to make changes to the data in the set; say yes to have direct access, and no for indirect access.

Note that up to answering of the questions for the YARDS set you can abort the creation process by responding with a "^" command. Also note that the "PARAMS" set refers to the contents of the Command System parameter and list menus (i.e. to the relations stored in the .mnurel HP file group).

Once you have made your responses for the last set, the scenario system begins construction of the new scenario. It makes up the cross reference list of SCENARIO field values by relation, and makes copies of the data in any relations for which you specified direct access (some copies will always be necessary, since each scenario always has its own PARAMS set). The copying process is time-consuming because of the large volume of data which must be manipulated.

ALIAS assumes you want to use the new scenario immediately and thus makes it your current scenario upon completion of the creation process.

Figure 6-4. Scenario Creation Menu, Stage 2

```
                                          Current scenario is
Scenario creation options include:
 ?      provides help                   | ^      exits scenario choice system
 @      lists existing scenarios        |      re-displays this menu
 name   specifies the name of the       | @name lists the composition
         scenario to take data from     |        of the 'name' scenario
         for a particular DB family.    |
--------------------------------------------------------------------------------
                           SCENARIO CREATION—STAGE 2

Please give the name of the scenario to take data for group CURRJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n

Please give the name of the scenario to take data for group DESCJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? y

Please give the name of the scenario to take data for group HISTJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n

Please give the name of the scenario to take data for group MISCJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n

Please give the name of the scenario to take data for group PARAMS   from.
COMMAND or NAME: fixit

Please give the name of the scenario to take data for group PROJJ    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? y

Please give the name of the scenario to take data for group YARDS    from.
COMMAND or NAME: fixit

Will you want to make any changes to the data in this group? n
```

6.1.3 Scenario Deletion

Choosing option 3 of the Command System's scenario menu brings up the menu shown in Figure 6-5. Here you are expected to give the name of the scenario you want to delete. It must be one you created yourself (unless you are a DBA-level user), and no one can be using it when you make the deletion request. If these conditions are met ALIAS will go ahead and perform the deletion. Like creation, it will be time-consuming. ALIAS must look through every data base relation in which the scenario had direct-access data, making sure it all is deleted.

Note that if some other scenario was set up to make indirect use of data in the one being deleted, this data will be transferred to the other scenario rather than being destroyed. Likewise, data which the deleted scenario was using indirectly will be unaffected by the deletion.

The scenario will be gone once you have deleted it, so you should do so only when you are sure you are through with its contents.

6.1.4 Scenario Structure Modification

Scenario structure modification changes the map that a scenario uses to find its data in central data base relations, thus effectively changing its contents. Five types of modification are possible:

1) Conversion of data in a relation or set from indirect-access to direct-access status. If you are using yard data from MAIN and find yourself needing to make yard data changes, you can have a copy of MAIN's yard data made which will belong to the scenario you are working with. You will then be responsible for keeping the yard data up to date, of course.

2) Changing from indirect access use of one scenario's data to indirect access use of another's.

3) Changing to a clean slate in some relation or set of relations. If the scenario you make this change to had its own direct-access data in those relations, it will be deleted.

Figure 6-5. Scenario Deletion Menu

```
                                              Current scenario is
Scenario deletion options include:
 ?      provides help                    |  ^    exits scenario choice system
 @      lists existing scenarios         |  +    moves you into scenario
 @name  lists the composition            |        creation menu
          of the 'name' scenario         |       re-displays this menu
 name   makes the 'name' scenario        |
        your current scenario            |
-------------------------------------------------------------------------------
                            SCENARIO DELETION MENU
Name of scenario to delete, or Command character:
```

4) Deletion of a scenario's direct access data in some relation or set of relations, and replacement of it with a copy of the data in some other existing scenario. In effect, the creation process is reproduced for selected relations.

5) Deletion of a scenario's direct access data in one or more relations, and "replacement" of it with indirect access use of the data in some other existing scenario.

You can make these changes on a relation-by-relation basis as well as to entire sets of relations (the same sets used during creation).

Choose option 6 on the Command System's scenario menu when you want to perform a modification. The menu shown in Figure 6-6 will result. You are expected to give the name of the scenario you want to modify. If you have change privileges for it, you will be passed on to the menu in Figure 6-7. Here your principal task is to choose the level of detail at which you will make modifications. If you give the CS command (Change Set) then you will make your modifications on the same sets of relations which were involved in the creation process. If you give the CR (Change Relation) command, then changes can be made on a relation-by-relation basis. The LS and LR (List Set/Relation) commands list the current structure of the scenario you are changing by showing the map of scenario key field values for each set or relation.

After you choose one change mode or the other, the menu shown in Figure 6-8 will appear. This is the "workhorse" menu of the modification subsystem, in which you specify the changes you want made using the syntax "set_name=new_source_scenario" or "relation_name=new_source_scenario". A list of the current sets and relations (whose names you must know exactly) is provided in Table 6-1 for your convenience. After you give one of these commands (with valid names on both sides of the "=" sign) you will be prompted with the usual "Will you want to change this data?" sort of question. An answer of "yes" causes a copy to be

Figure 6-6. Scenario Modification Menu, Stage 1

```
Scenario modification options include:
 ?      provides help                 |  ^      exits scenario choice system
 @      lists existing scenarios      |       re-displays this menu
 name   specifies the name of the     | @name lists the composition
         scenario to modify           |         of the 'name' scenario
_______________________________________________________________________________
                        CHANGE SCENARIO MAKEUP--STAGE 1
NAME of scenario to modify, or COMMAND: DEMO
```

Figure 6-7. Scenario Modification Menu, Stage 2

```
                  *** MODIFYING MAKEUP OF SCENARIO DEMO ***
Scenario modification options include:
 ?      provides help                 |  ^      exits scenario choice system
 CS     change composition by family  |  LS     list composition by family
 CR     change composition by file    |  LF     list composition by file
       re-displays this menu          |
_______________________________________________________________________________
                        CHANGE SCENARIO MAKEUP--STAGE 2
 COMMAND: CS
```

Figure 6-8. Scenario Modification Menu, Stage 3

```
Scenario modification options include:
 ?      provides help                 | ^      exits scenario choice system
 @      lists current scenarios       | @name lists composition of NAME
       re-displays this menu          | FAMILY=NEW SOURCE specifies change
--------------------------------------------------------------------------
                    CHANGE SCENARIO MAKEUP--STAGE 3
DB_FAMILY=NEW_SOURCE_SCENARIO, or COMMAND: MISCJ=FIXIT

Do you want a copy of that data made (no=only read it)? Y

Change made.
DB_FAMILY=NEW_SOURCE_SCENARIO, or COMMAND: ^
```

made and direct-access status to be given to the changed scenario for the data in the given set/relation, while "no" yields no copy and indirect access.

Use the following combinations of responses to implement each of the five kinds of change discussed above:

1) Indirect-to-direct, same data as before.
   Command: relation/set_name=source_scenario_name
   Response: Yes
   Note that the source_scenario_name is the same one listed for the given relation/set by using the LS or LR commands, since you are only changing the access status and not the contents of the scenario in this case. The difference between this and the creation syntax is the "yes" response.

2) Indirect use of A -to- indirect use of B.
   Command: relation/set_name=new_source_scenario_name
   Response: No

3) Wipe the slate clean.
   Command: relation/set_name=this_scenario_name
   Response: Yes
   This will delete any data belonging to this_scenario in the given relation or set, so use with care.

4) Direct -to- direct, different data.
   Command: relation/set_name=new_source_scenario
   Response: Yes
   This will delete the current data for this_scenario and replace it with a copy of that from new_source_scenario

5) Direct -to- indirect, different data.
   Command: relation/set_name=new_source_scenario
   Response: No
   Like 4 but no copy is made.

When you are finished making changes, or just want to review your work using LS or LR, give the "^" command to return to the second modification menu.

6.1.5 Listing Existing Scenarios

You can find out the names of existing scenarios by giving options 3 or 4 on the Command System's scenario menu. Option 3 types them on your terminal, along with their descriptions, while option 4 sends the list to the printer which is "current" on the

User Environment Parameter menu for your current scenario. This printed list can optionally include the structure of each scenario.

6.1.6 Auxiliary Scenario System Commands

Each scenario system menu shown in Figures 6-2 through 6-8 has a list of auxiliary commands at the top, commands which can be given to aid your main decision in the menu. Some of these perform the same function as they do in the Command System: "^" returns you to the last menu you were looking at, "&" retypes the menu, "?" causes some help to be printed. The less familiar ones and their functions are:

- **+** Puts you in the scenario creation subsystem; identical to choosing option 2 on the Command System scenario menu.
- **@** Lists the names of existing scenarios and their descriptions. Identical to option 3 on the Command System scenario menu.
- **@name** Lists the composition of the "name" scenario, at either the relation or the set level. Essentially lists the internal map which gives the SCENARIO field value for each relation/set for the "name" scenario.

6.2 BASIC STRATEGY FOR USING THE SCENARIO CAPABILITY

The scenario system provides you with a powerful set of tools for conducting analyses, but the ways in which these tools can fruitfully be used may not be completely clear. This section will provide a few rules of thumb and examples.

There are two main rules of thumb for scenario management. The first is that you should create a new scenario whenever you come to a point in an analysis where you are about to try some major changes which you think you might have to reverse. By trying the changes on a copy, rather than on the scenario you have painstakingly worked up, you can get back to your pre-change point instantaneously. And if you like the changes, you can always delete the pre-change scenario.

The overhead involved in creating such scenarios need not be time-consuming if you follow the second rule of thumb, which is to always choose indirect access to source scenario data (answer "no" to the "Will you want to make changes to this data?" question) over direct access unless you are sure you will need to make changes. Most of the time burden in scenario creation lies in the making of copies of source scenario data to permit direct access use; by minimizing the number of such copies you will minimize the amount of time required. You can always use the modification capability later to convert a relation or set from indirect to direct access status if you need to make changes.

Many studies focus on manipulation of only one part of the data base (such as projected SCN jobs), and are passive users of the rest. If your study falls in this category you need only set up direct access to the relations in the data base set that you are working with intensively.

You can use the scenario modification processor's "clean slate" option (method 3) to speed your work in some cases. For example, if you decide you must generate a completely new set of projected schedules for a scenario, you could go into the assigner, delete all the old assignments, and type in the new ones. A faster method would use the modification subsystem to delete all the records for your scenario in relation ncjodat.projj; then you would only need to type in the new ones in the assigner.

Suppose you have a study in which you are varying two major things, perhaps the number of shipyards and the structure of the projected SCN program. Suppose you have six scenarios for the study, having three different sets of yards and three different sets of schedules, and you want to combine these in various ways. The best method of accomplishing this is to create a new working scenario which indirectly accesses the data in one or more of the six scenarios you have. Then, to mix and match, you need only use the modification subsystem to change from indirect access to

scenario A's schedules to indirect access to scenario B's, and so forth. Since no data deletion or copying is involved, almost no time is consumed implementing the change in structure.

6.3 SCENARIOS AND DATA BASE QUERIES

It is important to keep the structure of a scenario in mind when you use RELATE to make queries involving its data. Since the scenario field appears in every central data base relation, you must specify the proper field value in your query in order to get responses which use only the data for the scenario you are interested in. And this field value is not always the name of your scenario; if it was making indirect use of some other scenario's data in a given relation, that other scenario's name must be specified in the query.

Suppose you want to know the state in which each projected SCN ship will be built. To do this, you need to combine information in the ncjodat.projj schedule relation and in the yardid.yards yard relation. Ncjodat.projj has the yard each ship is to be built at; its name can be used to reference into the yardid.yards relation to find the abbreviation for the state it is located in. Further suppose that the scenario you are using is called FYDP, and that FYDP has no yard data of its own, using that belonging to MAIN instead. Then the query would have to read:

```
OPEN FILE YARDID.YARDS;MODE=SHARED
OPEN FILE NCJODAT.PROJJ;MODE=SHARED
SELECT CLASS,HULL,AWARD,YARD,YARDID.STATE BY STATE,AWARD,CLASS &
 WHERE NCJODAT.SCENARIO="FYDP" AND YARDID.SCENARIO="MAIN" AND &
       NCJODAT.YARD=YARDID.YARD
PRINT
```

If you were not aware that FYDP was using MAIN's yard data indirectly, you might have specified NCJODAT.SCENARIO= YARDID.SCENARIO AND NCJODAT.SCENARIO="FYDP", in which case you

would have gotten no result, since there were no records in the yardid.yards relation with a SCENARIO field value of "FYDP". It is a good idea to use option 4 of the Command System's scenario menu to print a detailed list of the structure of a scenario before you begin to make queries about it using RELATE. When you do work with the DBU or any other module accessed via the Command System these details are handled for you automatically.

7.0 USING THE DATA BASE UPDATING SYSTEM

The Data Base Updating System (DBU) is your primary tool for making additions and changes to the data base. It is a different sort of environment (see Section 2) than most of the rest of ALIAS. Its primary displays are fill-in-the-blank screens which allow you to look at and/or change each field of an individual data record from a relation.

It provides many other services associated with the its basic mission of DB update support. This section will describe its commands and constituent parts in detail.

7.1 EXECUTING THE DBU

The DBU is accessed by choosing option 3 on the Command System's "TOP" menu. It has a very long startup lead time, but is somewhat different from other ALIAS modules in that this is a one-time cost. After you use it once in a Command System session and exit it, it remains "on hold". The next time you want it in the same session (just choose option 3 again) it will come up almost immediately. The long initial delay occurs because it needs to connect to most of the data dictionary and all of the legal values reference library, both of which are substantial data bases in their own right.

If the DBU produces garbled displays with many strange characters, this is because ALIAS has made an incorrect guess concerning your terminal type. As soon as the display is stable (meaning that the DBU is awaiting your command), type E and hit the return key. This will return you to ALIAS, where you must go to the User Environment Parameters menu and set the terminal type parameter to the correct value.

7.2 DBU SCREEN TYPES

You will encounter four kinds of screen while using the DBU, exemplified in Figures 7-1 through 7-4 (bracketed areas in the figures are the "blanks" you would see on a video display).

Figure 7-1. Sample DBU Menu Screen

```
*** SCREEN SKED_MENU
*** LAYOUT
SCREEN IS: SKED_MENU                                    SCENARIO IS: DEMO
===============================================================================
 ? for help              ALIAS DATA BASE UPDATE SYSTEM              =NAME jumps
===================choose a screen to use by number or NAME===================

                          COMMAND:[                ]

    1.  PROJECTED  NEW CONSTRUCTION, CONVERSION AND REACTIVATION JOB SCHEDULES
    2.  PROJECTED  REPAIR  OVERHAUL, REFUELING, AND SLEP JOB SCHEDULES
    3.  CURRENT    NEW CONSTRUCTION, CONVERSION AND REACTIVATION JOB SCHEDULES
    4.  CURRENT    REPAIR  OVERHAUL, REFUELING, AND SLEP JOB SCHEDULES
    5.  HISTORICAL NEW CONSTRUCTION, CONVERSION AND REACTIVATION JOB SCHEDULES
    6.  HISTORICAL REPAIR  OVERHAUL, REFUELING, AND SLEP JOB SCHEDULES




===========================No data may be changed here=========================
     Please place a command or option number after COMAMND and press RETURN
```

Figure 7-2. Sample DBU Data Screen

```
SCREEN IS: CLASS_CHARS              LATEST  DATA        SCENARIO IS: DEMO
==============================================================================
 ?  for help               ALIAS DATA BASE UPDATE SYSTEM            =NAME jumps
=======================Ship Class Characteristics=============================

                              COMMAND:[                 ]

        Class Name: [          ]            Owner:  [          ]

   Name for Reports: [                    ]

                           SHIP SIZE                               SHIP LIFE

           Length Beam    Height  Draft  Displacement     Standard Service Time
Waterline: [    ] [    ]  [   ]                           After Delivery: [   ]
Overall:   [    ] [    ]  [   ]                           in Time Units: [      ]
At Launch:                        [   ]  [       ]        Data Source[         ]
Light:                            [   ]  [       ]        Data Date  [         ]
Full Load:                        [   ]  [       ]        Entry Date
                                                          Entry By
===============Your privileges in this screen are: inspect===================
               Place a command after COMMAND and press RETURN
```

Figure 7-3. Sample DBU Comment Screen

```
SCREEN IS: CLASS_CHAR_COMT       LATEST DATA          SCENARIO IS: DEMO
===============================================================================
ESC L/D inserts/deletes  ALIAS DATA BASE UPDATE SYSTEM              + or - to page
===================Comments for Ship Class Descriptions========================

                            COMMAND:[                ]
  Class: [          ]                                       Data Date  [          ]
                                                            Entry Date [          ]

                                   Comments
         [                                                        ]
         [                                                        ]
         [                                                        ]
         [                                                        ]
         [                                                        ]
         [                                                        ]
         [                                                        ]
         [                                                        ]
         [                                                        ]
                                                            Entry By
==================Your privileges in this screen are: inspect==================
  Changes to comments will be posted on finish with this item in data screen.
```

Figure 7-4. Sample DBU Help Screen

```
SCREEN IS: CLASS_MENU_HELP                          SCENARIO IS: DEMO
=========================================================================
 ? for more help      ALIAS DATA BASE UPDATE SYSTEM HELP     / to leave help
==============description of the purpose of the class menu===============

                     COMMAND:[                ]

   Before any information about a given ship class (or about ships of that
class) may be entered into the ALIAS data base, a basic description of the
class's ships must be entered using the CLASS_CHARS screen (option 1).

   If any projected ships are to be of the newly entered class, it is a good
idea to enter planning factors for the class (option 2).  Without planning
factors many ALIAS modules will not be able to operate properly.  Only when
you choose to enter specific, 'actual' data for each projected ship in a class
can the entry of planning factors be bypassed.


====================no data may be changed here==========================
          Please place a command after COMMAND and press RETURN
```

There are menu, data, comment, and help screens. Menu screens perform a function similar to that of Command System choice menus: they help you get around by telling you what your options are.

Data screens are the workhorses. There is one for each central data base relation (more or less). Each one has blanks which correspond to its relation's fields, and can thus display a single record from the relation at a time.

Comment screens are auxiliaries of data screens (there is one for each data screen). They let you enter textual comments about the data values in the record you are currently working with in the data screen.

Help screens display text which tries to explain where you are and what you can do. Each data screen has its own dedicated help screen, and there are a large number of general-purpose help screens which describe every aspect of DBU use.

There are some subtle differences in the operation of menu and data screens on the one hand and help and comment screens on the other which place minor limitations on the commands you can give in the latter. These limitations will be covered below.

7.3 KINDS OF COMMANDS

Like all interactive ALIAS processors, you must give the DBU commands to tell it want you want done. There are two basic types of commands: those that you place in the COMMAND field (located at the top of each screen; see Figures 7-1 through 7-4) and which have to do with the whole screen in some sense; and those which have to do with only one field on the screen, which you give by placing the cursor on the field and hitting the Escape key.

COMMAND field commands are generally a single character (e.g. "Q" for "quit"); you place them in the field and hit the RETURN key and the DBU executes them. These commands do one of three things: cause a different screen to be displayed, change a DBU mode setting, or alter/save the data record showing in the "blanks".

Escape commands are also single characters, which you type in response to a prompt given after you hit the Escape key. These will either change the contents of the field the cursor was at when you hit the Escape key, or else will cause some help about that field to be printed.

Figure 7-5 is the help screen which summarizes the commands available.

7.4 MOVING AROUND IN THE DBU

In order to perform the primary DBU function of modifying the data base, you must first get to the screen which handles the data you want to modify. There are three basic ways of doing this: following the menus, jumping directly to the screen, and popping back to it.

If you have just started the DBU, and don't know the name of the screen you need, the best means of finding it is by working your way down the hierarchical tree formed by the DBU's menu screens. These menus start at a high level and become more detailed; eventually you will pick an option on one which will bring up the data screen you want. For example, if you wanted to change schedule planning factor data and were looking at the MASTER menu shown in Figure 7-6, the best choice would be option 1 (ship classes), since planning factors are at the class level. This would bring up the CLASS_MENU screen, also shown in Figure 7-6. This menu has an option which matches what you want exactly (2); choosing it results in display of the SKED_PF_MENU screen,

Figure 7-5. On-Line Summary of DBU Commands

```
SCREEN IS:  COMMANDHELPL                             SCENARIO IS: DEMO
=======================================================================
 ?* for beginner's help ALIAS DATA BASE UPDATE SYSTEM HELP     / to leave help
=====================summary of command codes==========================
                             COMMAND: [              ]
 USE      TO MOVE TO                 USE        TO AFFECT FIELDS/ALTER DB
 ----     ------------------------   ----       --------------------------
 ?        help subsystem             N          bring next data item to screen
 ?NAME    help screen NAME           K          clear fields
 C        comment screen             S          search for matching data item
 /        top screen or leave help   B          negate search/rewind file
 ^        previous screen            A          add screen data to DB
 =        next screen                D          delete screen data from DB
 =NAME    jump to screen NAME        U          update DB, alter data date
 Q        main menu system           M          modify DB, same data date
 :        MPE command mode           V          verify data in fields ok
 T or TH  TDP or HP editors          ESC ←      next legal value (fld=cursor)
 L        set INSPECTion mode        P          print data in screen set
 ESC R    recalculate dates          ←          page forward/back, this item
 ESC ?    help for field             ESC I or D  insert/delete lines (cmts)
=====================no data may be changed here=======================
   Use ?x, where 'x' is one of the command codes, to find out more about 'x'.
```

Figure 7-6. DBU Menus: MASTER and CLASS_MENU

```
SCREEN IS: MASTER                                               SCENARIO IS: DEMO
=================================================================================
 ?  for help             ALIAS DATA BASE UPDATE SYSTEM                =NAME jumps
================choose a screen to use by number or NAME=========================

                         COMMAND:[                ]

                         1. SHIP CLASSES
                         2. SHIP JOB TYPES
                         3. SHIP JOB SCHEDULES
                         4. SHIP YARDS
                         5. SHIP DEACTIVATIONS





======================No data may be changed here================================
    Please place a command or option number after COMAMND and press RETURN
```

```
SCREEN IS: CLASS_MENU                                           SCENARIO IS: DEMO
=================================================================================
 ?  for help             ALIAS DATA BASE UPDATE SYSTEM                =NAME jumps
================choose a screen to use by number or NAME=========================

                         COMMAND:[                ]

                         1. CLASS CHARACTERISTICS
                         2. SCHEDULE PLANNING FACTORS





======================No data may be changed here================================
    Please place a command or option number after COMAMND and press RETURN
```

in which you would choose among different kinds of planning factors, finally arriving at a data screen.

This process is reliable but somewhat tedious, especially after you have some experience with the DBU. If you know that the name of the screen you want to use is NC_SKED_PF, but you are in the MASTER menu, you can still have it displayed immediately by giving the command "=NC_SKED_PF". You can give the jump ("=") command in any data or menu screen, and can specify the name of any other data or menu screen. Table 7-1 lists all DBU screens by category, serving as a reference for jumping.

The "=" command works only for jumps to menu and data screens. To jump to a particular help screen, you must use the command "?HELP_SCREEN_NAME".

You can also change screens by "backing up". If you were last in the NC_SKED_PF screen, but are now in the PROJ_NC_SKED screen, you can go back to NC_SKED_PF just by giving the "^" command. The meaning is thus similar to the "^" command of the Command System. The DBU keeps a history of the last eight menu and data screens you have used, so you can back up eight times (there is unlimited backup for help and comment screens). If you try to back up more than eight times you are always left in the MASTER menu.

As in the Command System, the "/" command will "pop you to the top". The "top" is defined as the MASTER menu if the command is given in a menu or data screen, and the last-used menu or data screen if you are in a comment or help screen.

To leave the DBU, give the "Q" (quit) command in any menu or data screen. The "E" command will also get you out of the DBU, but will terminate the DBU process, forcing you to pay the initialization time penalty the next time you want to use it.

TABLE 7-1. Annotated List of DBU Screens By Type

SCREEN	PURPOSE
	COMMENT INSPECTION/ENTRY SCREENS
CLASS_CHAR_COMT CURR_NC_SKED_C CURR_RE_SKED_C DEACTIVATE_COMT HIST_NC_SKED_C HIST_RE_SKED_C NC_SKED_PF_COMT PROJ_NC_SKED_C PROJ_RE_SKED_C YARD_CHARS_COMT	Each of these screens is associated with a data screen, and is intended for entry and modification of comments about the data displayed on the associated screen. Comments are always associated with a particular record in the associated data relation.
	DATA ENTRY/MODIFICATION SCREENS
CLASS_CHARS CURR_NC_SKED CURR_RE_SKED DEACTIVATE HIST_NC_SKED HIST_RE_SKED NC_JOB_TYPES NC_SKED_PF PROJ_NC_SKED PROJ_RE_SKED RE_JOB_TYPES YARD_CHARS	These are the DBU's data entry screens, the reasons for its existence. Names are mnemonic, with SKED indicating schedules, HIST, CURR, and PROJ referring to historical, current, and projected epochs respectively, NC and RE to new construction type jobs (reacts and conversions too) and to repair-type jobs (includes SLEPs). CHARS stands for characteristics, PF for planning factors.
	HELP SCREENS WHICH TELL ABOUT A MENU OR DATA SCREEN
CLASS_CHAR_HELP CLASS_MENU_HELP C_NC_SKED_HELP C_RE_SKED_HELP DEACTIVATE_HELP H_NC_SKED_HELP H_RE_SKED_HELP NC_SKED_PF_HELP P_NC_SKED_HELP P_RE_SKED_HELP SKED_MENU_HELP SKED_PF_HELP YARD_CHARS_HELP YARD_MENU_HELP JOB_TYPE_HELP	These help screens have text which talks about "their" data or menu screen. They are displayed when you enter a "?" in that screen. The names are mnemonic and similar to those of the associated data/menu screens. H, C, and P stand for HIST, CURR, and PROJ respectively.

TABLE 7-1. Annotated List of DBU Screens By Type

SCREEN	PURPOSE
	HELP SCREENS WHICH DESCRIBE OVERALL DBU OPERATION
BACKHELP	How to retrace your steps through the screens you have been in already.
CHAINHELP	How to use menu screens to find your way around.
COMMANDHELP	General discussion of types of DBU commands.
COMMANDHELPL	Table of DBU command code characters.
CURSORHELP	General discussion of the screen cursor and how to move it around the screen.
CURSORHELPL	Table of control codes for cursor movement.
DCHECKHELP	General discussion of data validation which is done.
DELETEHELP	What happens when you try and delete data.
HELPHELP	How to use the help subsystem.
IFIELDHELP	Further discussion of screen enhancements.
JOINHELP JOINHELP2	Why and how join checking is done as part of the validation process.
JUMPHELP	How to have the DBU display the next screen you want to use without going through the menus.
LEGALHELP	Why and how legal value checking is done as part of the validation process.
MAINHELP	General introduction to DBU and to the system-level help screens.
MOVEHELP	General discussion of how to move from screen to screen.
NEXTHELP	How to move to the default next screen.
PUSHHELP	How help and comment screens are entered and exited.
READHELP	General discussion of how to interpret the display enhancements used for screen fields.
SLOWHELP	Why is this !!!! thing so slow, and inconsistently?
	MENU SCREENS
CLASS_MENU JOB_TYPE_MENU MASTER SKED_MENU SKED_PF_MENU YARD_MENU	These screens guide you to the proper data screen for your purposes.

DBU security may prevent you from seeing a particular screen. See the Data Base Administrator if you feel this is a mistake.

7.5 ON-LINE HELP

The DBU has very extensive on-line help to aid you at the terminal. It is organized in the form of a help "subsystem" consisting of a large number of text screens. You can activate the subsystem at any time by giving the "?" command. The result will be a help screen dedicated to the menu or data screen you were in, describing its purpose and operation. It is a good idea to read the help screen any time you use a new data screen, since it will inform you of any unusual features.

If you require additional help, you can give the "?" command again in the help screen. Help screens are linked together in a chain which you can traverse by repeated issuance of the "?" command. To exit the chain without popping back through all the help screens you have read, just issue the "/" command and you will be back in the menu or data screen.

As noted in the previous section, you can also jump directly to any help screen by tacking its name onto the "?" (e.g., "?HELPHELP"). One of the most commonly used help screens is the summary of available commands (shown in Figure 7-5), whose name is "COMMANDHELPL".

Help is available about individual fields on a data screen as well as about the screen as a whole. To find out what a field is for, move the cursor to it, hit the Escape key, and give the "?" Escape command. This will bring up a special field-help screen which will query the data dictionary about the field and display the results for your perusal.

7.6 USING DATA SCREENS

7.6.1 Reading and Working With the Display

When you have a data screen showing on your display, the first thing you will need to do is decode its format. There will be a title just under the "ALIAS DATA BASE UPDATE SYSTEM" header, and a hint at the bottom, which will give you some indication of what the screen is for and how to use it. There will be a COMMAND field, showing in inverse video, just under the title. This is where you put your command characters before hitting the carriage return key.

Embedded in the line of equal signs at the bottom of the screen will be an indication of your data change privileges in the screen. If any of Inspect, Add, Modify, Update, or Delete do not appear in the list then you will not be allowed to give the corresponding command. A lack of privileges can result from denial by the DBA. A more likely cause when you find yourself with Inspect privileges only is the structure of the scenario you are using: if the scenario indirectly accesses the data belonging to another scenario in the relation your current data screen serves, you will naturally not be allowed to make any changes. If you have a need to make changes, you must first alter the structure of the scenario.

There will be a number of labeled blanks or windows. These are the areas where field values from the data base will be displayed. There are four kinds of window; you can distinguish them by their differing screen enhancements. Those which are in inverse video (bright green or white) are "ordinary" fields, meaning that you can put whatever value you please there (except you can't put words in a date field or a numeric field).

Those which are underlined are "legals" fields, meaning that whatever value is there must match one on a relatively short list of legal values. You can flip through this list for any

legals field by placing the cursor on the field, hitting the Escape key, and giving the + or - command to go forward or backward through the list.

Those which are both underlined and in inverse video are "key" fields. These are typically part of the unique key field set for the relation the screen is servicing. In order to make additions or modifications to the contents of the relation, you will have to put values in these fields which are consistent with the rest of your scenario. For example, you cannot put "XYZ" in the "yard" window of a schedule screen unless there is data for a yard named "XYZ" in the yardid.yards relation for your scenario.

Some windows will have no enhancement at all, such as the entry date field which appears at the bottom right of most screens. This means that you cannot change the value of the field; its values will be set automatically by the DBU.

The DBU's data validation logic will check the values in the legal and key fields only when you give a command which would change the contents of the data base.

Once you understand the layout of the screen, you will need to be able to move the display's cursor (the blinking white or black box or underbar) from field to field in order to make changes. The TAB key is the primary means of doing this; hitting it jumps the cursor to the next field (to the right, or to the first one on the next line). Within a field the BACKSPACE key will perform its normal function, as will the space bar.

The cursor control keys on your keyboard (arrow keys) will not work as they usually do. However, if you are using an HP terminal, you will notice that the function keys are labeled as performing actions like "left" and "up". The function keys can be used in place of the cursor control keys. If you are not using an HP terminal, control-R moves the cursor right one space,

control-L moves it left, control-U moves it to the next field in the upward direction, control-D moves it down, and control-Z will redraw the display if it becomes garbled somehow.

7.6.2 Adding Data

To add a new record to the data base, just fill in all the fields on the screen with the proper values, put the "A" command into the COMMAND field, and press the return key. The DBU will validate the data and do the addition. If the data does not pass the validation check the DBU will tell you why. You can make validation more likely by using the ESC +/- commands to set legal field values rather than trying to set them from memory.

7.6.3 Finding Data

Often you will want the DBU to display the field values of a particular data record, either for your inspection or so you can modify them. There are two basic ways of doing this. First, you can use the "N" (next-record) command to look through the underlying data relation one record at a time until you come to the one you are interested in. This is fine if the record is near the top in the sort ordering (records are displayed in the sort order of the key fields), but quite tedious if it is necessary to look through hundreds to find one.

The alternative is the "S" (search) command. To use it, take the following steps:

1) Give the "K" (clear) command to erase all the fields on the screen. It is important not to omit this step, as it is the only way of erasing the entry_by and entry_date fields which appear on most screens.

2) Fill in selected fields with "target" values, values which the record you are looking for must match. If a field is of the numeric or date variety the DBU will demand that records match the value you specify exactly. If it is an alphanumeric field, the DBU will demand only that there be a match on the characters you specify. Suppose you are looking for schedules for the DDG-51 class; you should put "DDG-51" in the class field on the given schedule screen and give the "S" command.

AD-A150 424 ACQUISITION AND LOGISTICS INFORMATION AND ANALYSIS SYSTEM (ALIAS) USER'S GUIDE(U) DECISION-SCIENCE APPLICATIONS INC ARLINGTON VA M S CAREY ET AL. 31 OCT 84 DSA-618 N00014-82-C-0813

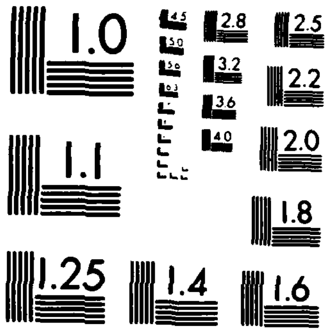
4/4

UNCLASSIFIED F/G 15/5 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

But if you were interested in schedules for any DDG class, you could just type "DDG" in the class field and the DBU would return records with any of DDG-2, DDG-51, etc. in their class field. However, if you put a number in the hull field (which is numeric), only records with an exact match on that number will be returned ("1000" would not be returned in response to a specification of "1").

The more fields you fill in, the more precise your targeting will be. However, if you make a mistake in specifying a value, the DBU may claim that it cannot find a match (and it truly can't), when what you are really looking for does in fact exist. Thus, if you are unsure about a value, leave that field blank.

3) Put the "S" command in the command field and hit return. The DBU will conduct the search and place the first matching record it finds onto the display. If there are additional matching records, you can look through them by giving the "N" command in the usual way.

4) When you are through with a given search and want to "undo" it, you may either specify a new search (using the K and S commands), or you can clear all searches and return to the beginning of the file by giving the "B" command.

In addition to the search capability, there is also a more general inspection-mode control capability. As noted above, most central data base relations can hold a history of records for a given data item. The DATADATE field is part of the unique key field set for these relations; when new data come in and an update is required, it can be added with a later data date instead of being written over the record that is out of date. This makes it possible to perform reconstructions, but the presence of many records for each item can be a hindrance for day-to-day operations, when generally only the latest data is of interest.

A similar problem revolves around the distinction between actual and projected data. For example, the current SCN job schedule relation has a DATETYPE field, in which the nature of a data record can be specified. For query purposes, often only actual or only projected data will be of interest.

The nature of the records which the "N" or "S" commands offer up is determined by the DBU's inspect-mode setting. The current setting is shown in the middle of the very top line of the display for every data screen (just to the left of the current scenario name). The setting will take the form of "LATEST PROJECTED DATES", "ALL DATES", "LATEST DATA", etc., depending on the nature of the screen. To change the setting, give the "L" command. You will be prompted for a data date restriction setting (LATEST or ALL) and a date type restriction setting (ACTUAL, PROJECTED, or ALL). If you choose LATEST, only the records with the most-current data date for each item will be returned by Searches and Nexts. ALL will cause all records to be returned.

Note that if the date type setting is ALL and the data date restriction LATEST (the mode setting display will read "LATEST DATES"), and there are both actual and projected records for a given item (e.g. ship), the DBU will display only the record with the latest data date, not both the latest actual and the latest projected one.

7.6.4 Changing a Data Base Record

Having retrieved a record onto the screen, you are free to change its values. There are two ways of posting your changes to the data base. The first, giving the "M" (modify) command, will cause the DBU to replace the original record with one with the values now displayed. Modification can work only if you have not changed any key field values, because the DBU must use these values to find its way back to the original record. If you have changed key field values other than the DATADATE field, you should use the A (add) command, since you have effectively created a new data item.

If you wish to retain the old data record for historical or audit-trail purposes, you can still put the altered record into the data base using the "U" (update) command. This is similar to an add command in that it adds a new record to the relation

without disturbing the one you retrieved, but differs in the validation checks performed. Update insists that there already be a record in the relation with key field values identical to all those showing on the screen EXCEPT for DATADATE, which you must have assigned a new value to.

As a general rule, most data entry changes should be posted using Update, while corrections of data entry errors should be posted using Modify.

7.6.5 Deleting a Data Record

You can remove a data record from the data base, after having retrieved it onto the screen, by issuing the "D" command. Be aware that this can lead to deletion of more than just the record you see. If there is data elsewhere in the data base which is "subsidiary" to it, the DBU will delete that as well (after prompting you for approval to go ahead). If, for example, you ask that the DDG-51 record be deleted in the class characteristics screen, this is tantamount to a request that all information having to do with DDG-51's be flushed from your scenario.

Note that this step is taken only when you are deleting the last record for the given item. If, for example, you are only deleting outdated records for the DDG-51 class, leaving at least one (say the most current one) in place, then no wholesale deletion will occur.

7.6.6 Validation

Data record validation occurs whenever you give the "A", "M", "U", or "V" commands. "V" (verify) performs only the validation, allowing you to fill in only the key and legals fields on a data screen and then see if they will be approved before you go on to fill in the rest of the fields.

Three types of validation are typically performed. First, the DBU checks to see if the values of the key fields (other than

DATADATE) which you have filled in form a unique set for your current scenario. They must be unique if you have issued the "A" command (i.e. the given item must not have been previously added), and must NOT be unique otherwise.

Second, a check is made to make sure that companion data is present elsewhere in the data base. This is called a "join check", because it ensures that you will be able to make joins among relations when you make queries using RELATE. The DBU consults the join requirement specifications given in the filjoin.db relation in the data dictionary in deciding what to check. An example would be the checks made for a projected SCN job schedule record: a match would be required on the class name in shdesc.miscj and on the yard name in yardid.yards. A match on class and job type in the ncjdat.descj schedule planning factors relation would be "recommended."

The basic principles from which the join requirement specifications were derived are as follows:

1) All values in a CLASS field in any relation must have a match in the same scenario in the shlife.miscj, shdesc.miscj, and valcls.mnurel relations.

2) All values in a YARD field in any relation must have a match in the same scenario in the yardid.yards and valyds.mnurel relations.

3) All values in a NCJOBT field in any relation must have a match in ncjobt.legals, jobtyp.legals, and in the same scenario in vljtyp.mnurel.

4) All values in a REJOBT field in any relation must have a match in rejobt.legals, jobtyp.legals, and in the same scenario in vljtyp.mnurel and vlrjob.mnurel.

5) The set of values for the SCENARIO,CLASS,HULL,COMNUM fields in any record in the deact.miscj relation must have a match in <u>one of</u> ncjodat.histj, ncjodat.currj, or ncjodat.projj (you can't retire a ship that was never commissioned).

6) The set of values for the SCENARIO,CLASS,NCJOBT fields in any ncjodat.projj record should have a match in the ncjdat.descj relation (there should be schedule planning

factors for every scheduled SCN job), but this is not an absolute requirement.

7) Every record in a relation holding comment text must be associated with a record in its companion data relation, the association being by identical values in all the fields forming the data relation's unique key field set.

7.7 COMMENTING

The DBU makes it possible to keep notes or comments concerning every data record displayable on a DBU data screen. Each data screen has an associated comment screen which can be called up by giving the "C" command. The comment screen displays (without permitting changes to) the current values of the data screen's unique key field set, and provides a block of 9 65-space lines for typing of textual notes. These notes are saved in a comment relation associated with the data relation(s) (meaning the comment relation has a similar name, is stored in the same file group, and has the same unique key field set as the data relation, with the addition of a line number field).

Existing comments are retrieved automatically for the current record when the comment screen comes up. They can be modified freely.

The DBU's capacity is 24 lines of comments per data record, although the window on comment screens is only 9 lines long. Comment screens will respond to the "+" and "-" paging commands in much the same was as the assigner does, allowing you to peruse and alter long comments a little at a time.

Two special Escape commands are available to support comment text modification: Escape I inserts a blank line into the text at the point on the window where you have placed the cursor, while Escape D deletes the line at the current cursor position and closes up the text.

The update mechanism for comments differs from that for ordinary data records. The "A", "U", and "M" commands do not operate in comment screens---instead, comment text is retained in the DBU's memory when you return to the data screen from the comment screen, and is posted to the comment relation when you finish with the data record showing on the data screen. "Finishing" can be triggered by any command which would show another data record, or any command which would post the data record to its relation.

There are a few subtle restrictions on this capability. The first is that changes to key field values will cause any comment changes you have made to be lost (much like the regular Modify command, the DBU will not be able to find the original comments in order to change them). For this reason, you should always fill in all key fields before calling up the comment screen, and should use the Verify command to make sure they will be accepted.

7.8 PRINTING

The DBU provides a data printing service which is accessible from any data screen by issuance of the "P" command. The screen which responds will list (and execute at your command) any special report generators which have been designed for the given data screen's underlying relation. Often there will not be any of these, but the screen also will cause a formatted PRINT command to be issued for the contents of the relation, with the contents being restricted both by any Search command you have executed and by the display mode subsystem's setting. Thus, to get a printout of the latest actual schedules for current destroyer construction jobs, you could go to the CURR_NC_SKED screen, set the inspect mode to LATEST ACTUAL DATES, do a Search specifying that CLASS field values must begin with "DD", give the "P" command, and choose option 1. The last two steps can be given in shorter form by giving the command "P1" in the data screen.

The print screen asks which printer to send the results to, offering the SEA 90 daisy wheel, SEA 90 dot matrix, PMS 392 line printer, and your terminal as options.

7.9 MISCELLANEOUS SERVICES

An MPE command can be given at any time by giving the ":" command and responding to the resulting prompt. The TDP and HP editors can be used by giving the "T" or "TH" commands, respectively.

A special service, schedule data recalculation, is available only in the PROJ_NC_SKED screen. It is designed to allow quick recomputation of schedule dates from a new basis date using the schedule planning factors for the current scenario stored in the ncjdat.descj relation. To use the service, type a date into any milestone date field on the screen, leave the cursor in that field, and give the Escape R command. The DBU will find the appropriate estimate of intervals between milestones in ncjdat using the values of the class, yard, job type, and series type fields on the screen (note that if no exact match is found on the yard name, intervals for yard "ANY" will be used for the given job if found in ncjdat) and will then compute a complete set of milestones just as the assigner would.

8.0 USING THE ASSIGNER

The assigner is a tool for creating and changing sets of projected ship construction schedules. It works by reading your scenario's schedules from the data base, converting them into a shipyard-assignments format, allowing you to make changes to the assignments, and then converting them back into schedules again.

This section will describe how to use the assigner and to some extent how it works. With regard to the latter, it is important to understand the methods the assigner uses to create new schedules in order to make full use of its capabilities.

To execute the assigner, move to the assigner choice menu in the Command System and choose option 2.

8.1 SCHEDULES

For the assigner's purposes, a schedule is a record in one of six central data base relations (files ncjodat and rejodat in each of groups histj, currj, and projj) which describes the nature and timing of a shipyard job. The "ncjodat" relations are for jobs of the new construction variety, defined as any which add ships to the Navy force level under a new class/hull number designation (thus including new construction, conversion, and reactivation), and also including commercial new construction and conversion. The "rejodat" relations are for jobs of the repair variety, defined as being done on a ship in the active force (thus including SLEPs and nuclear refuels as well as RO's, etc.). The three groups are for historical jobs (those completed), current jobs (those awarded but not yet delivered), and projected jobs (those not yet awarded).

The display produced by the assigner can contain jobs from any of the six relations, but the assigner is capable of creating schedules only for projected new contruction-type jobs (i.e. in the ncjodat.projj relation).

Figure 8-1 shows some schedule records from ncjodat.projj. Figure 8-2 expands the first record shown in Figure 8-1 into a more readable format. It is important to understand the structure of this record if you are to gain a full understanding of assigner operations. Its fields are:

1) $LINE: The ID number which RELATE assigns to each record.

2) SCENARIO: The name of the scenario the schedule belongs to (remember that other scenarios might use the record indirectly).

3) CLASS: The name of the ship class the job will be done for. A record for this scenario with this class name must appear in shdesc.miscj, shlife.miscj, and valcls.mnurel for data base internal consistency.

4) HULL: The hull number of the ship the job will be done on. This is a numeric variable.

5) COMNUM: Commissioning number. If a ship has been mothballed and then reactivated, it will have more than one job record in the three ncjodat.relations. The initial construction record will have a COMNUM field value of 1, while the first reactivation will have a value of 2, the second a value of three, etc. In combination with CLASS and HULL, this field uniquely identifies the job for force structure impact purposes.

6) YARD: Name of the shipyard the job is to be performed at. A record for this scenario with this name must appear in yardid.yards and valyds.mnurel.

7) NCJOBT: "New Construction JOB Type"---a code indicating the type of job to be performed. The code must be on the list of legal codes in ncjobt.legals and jobtyp.legals.

8) JSTYP: "Job Series TYPe"---a code indicating whether the job is of the lead, first follow, etc. variety. The code must be on the list of legal codes in jstyp.legals.

9) CUSTOMER: Name of the organization the job is being done for. Almost always "USN".

10) SHIPNAME: If known, the full name of the ship.

11) CMETHD: Construction method to be used. The code which appears here must appear on the list of legal codes in cmethd.legals.

Figure 8-1. Sample Schedules From Ncjodat.projj Relation

```
$LINE SCENARIO     CLASS      HULL COMNUM YARD      NCJOBT JSTYP  CUSTOMER SHIPNAME
CMETHD     APPROP
     AWARD      START       KEEL      LAUNCH  DELIVERY COMMISSION DAYSADDED ASNORDER
DATADATE DATASOURCE ENTRYBY ENTRYDATE AUTO
PROGVAR1 PROGVAR2 SUBRELUMAP

  334 DEMO          LSD-41        42      1 AVONDALE NEWCON ORDFOL USN
MODULZ 10/01/1985
11/01/1985 11/01/1986  5/01/1987  1/01/1989  5/01/1990  6/01/1990         0 57553221
10/28/1984 908/SHAPM  MARK      10/28/1984 YES
     42        0         0
  335 DEMO          LSD-41        43      1 AVONDALE NEWCON ORDFOL USN
MODULZ 10/01/1984
11/30/1984  6/30/1986                        1/30/1989                    0        0
9/01/1984 908/SHAPM  DBA        9/25/1984 YES
     41        1         0
  336 DEMO          LSD-41        44      1 AVONDALE NEWCON ORDFOL USN
MODULZ 10/01/1985
11/01/.985  7/02/1987  1/02/1988  9/02/1989  1/02/1991  2/02/1991         0 57553221
10/28/1984 908/SHAPM  MARK      10/28/1984 YES
     43        0         0
  337 DEMO          LSD-41        45      1 AVONDALE NEWCON ORDFOL USN
MODULZ 10/01/1986
11/01/1986  3/01/1988  9/01/1988  5/01/1990  9/01/1991 10/01/1991         0 57553221
10/28/1984 908/SHAPM  MARK      10/28/1984 YES
     44        0         0
  338 DEMO          LSD-49        49      1 AVONDALE NEWCON LEAD   USN
MODULZ 10/01/1987
11/01/1987 11/01/1988  7/01/1989  6/01/1991  6/01/1993  7/01/1993         0 57553222
10/28/1984 908/SHAPM  MARK      10/28/1984 YES
     49        1         0
  339 DEMO          LSD-49        50      1 AVONDALE NEWCON ORDFOL USN
MODULZ 10/01/1987
11/01/1987  5/02/1989  1/02/1990 12/02/1991 12/02/1993  1/02/1994         0 57553222
10/28/1984 908/SHAPM  MARK      10/28/1984 YES
     50        0         0
```

Figure 8-2. A Single Sample Schedule

FIELDNAME	VALUE
$LINE	334
SCENARIO	DEMO
CLASS	LSD-41
HULL	42
COMNUM	1
YARD	AVONDALE
NCJOBT	NEWCON
JSTYP	ORDFOL
CUSTOMER	USN
SHIPNAME	
CMETHD	MODULZ
APPROP	10/01/1985
AWARD	11/01/1985
START	11/01/1986
KEEL	5/01/1987
LAUNCH	1/01/1989
DELIVERY	5/01/1990
COMMISSION	6/01/1990
DAYSADDED	0
ASNORDER	57553221
DATADATE	10/28/1984
DATASOURCE	908/SHAPM
ENTRY_BY	MARK
ENTRY_DATE	10/28/1984
AUTOMOD	YES
PROGVAR1	0
PROGVAR2	0
SUBRELUMAP	0

12) APPROP: Projected date of appropriation of the ship (job).

13) AWARD: Projected award date.

14) START: Projected start date.

15) KEEL: Projected date the keel will be laid, or for land-level facilities, date of first major assembly area occupancy. Date of drydocking for reactivations and conversions.

16) LAUNCH: Date the ship will go into the water.

17) DELIVERY: Date the ship will be delivered to the customer.

18) COMMISSION: Date the ship will be commissioned. Does not apply to commercial jobs.

19) DAYSADDED: Days added to the ship's basic service life by the job. This will be 0 for new construction jobs, but reactivations may offer some of the life-extension benefits of a SLEP.

20) ASNORDER: A variable used by the assigner for sort-ordering, of no interest to you.

21) DATADATE: Date as of which the record's data is valid. If the record is generated by the assigner, this will be the same as the entry date.

22) DATASOURCE: Note of the source of the data.

23) ENTRY_BY: User name of the person who entered the data in the record.

24) ENTRY_DATE: Date the record was added to or updated in the relation.

25) AUTOMOD: "Yes" if the assigner may change the contents of the record during its update step. Setting this value to "no" after making changes to the record in the DBU ensures that your changes will not be thrown out by a subsequent assigner run.

26) PROGVAR1: Used by the assigner for computations.

27) PROGVAR2: Used by the assigner for computations.

28) SUBRELUMAP: Indicates to the assigner and DBU which relations subsidiary to ncjodat.projj have records associated with this record. The value will be 0 if there are none.

From the point of view of identification of the nature of the record's job, the most important fields are SCENARIO, CLASS, NCJOBT, and JSTYP. From the point of view of contents, the milestone date fields are of most interest. The assigner display contains the values of all the identification fields, and of one of the milestone fields, in some form for every job it shows.

8.2 READING THE ASSIGNER DISPLAY

When you execute the assigner, it will first "load" all the relevant schedules for your scenario, meaning it will read them and convert them into the assignments format. It will then type the first page of assignments onto your terminal display, which will look something like Figure 8-3.

A shipyard assignment is a ship-job projected for performance at a given yard in a given period. It is thus equivalent to a schedule, but it appears on the assigner display as a numeric value in one of the display cells. For example, the top left cell shown in Figure 8-3 indicates that there are two LSD-41's assigned to the Avondale yard.

The display is organized as a table of rows and columns: each row has the assignments for jobs of a particular type to be done on ships of a particular class at a given yard. Each column indicates the number of jobs to be performed in each period. The periods can be in units of days, weeks, months, quarters, calendar years, or fiscal years. The units for the sample are fiscal years, as indicated by the "Time in:" label at the top right of the page. There are two rows of columns headings: the top one just counts the number of periods starting at 1, while the one below it gives whatever period labels make sense, here the last two digits of the fiscal year the column represents.

The cell values are based on the projected number of occurrences of a basis milestone date, which can be any one of

Figure 8-3. A Typical Assigner Display Page

```
Scenario: DEMO               *SHIP ASSIGNMENTS*   Page  1A   Time in: FISCYR
Yard      Period:| 1  2  3  4  5  6  7  8  9                               |  9
   Shipclass   T|86 87 88 89 90 91 92 93 94                                |TOT
AVONDALE     #01|--+--+--+--+--+--+--+--+--|-------------------------------|---
  1 LSD-41      | 2  1                                                     |  3
  2 LSD-49      |      L2  2  2                                            |  6
  3 T-AO-187    | 2  2  2  2  2                                            | 10
BIW          #02|--+--+--+--+--+--+--+--+--|-------------------------------|---
  1 CG-47       | 1  1  1  1                                               |  4
  2 DDG-51      |   Y1 F2  1  2                                            |  6
EB GROT      #03|--+--+--+--+--+--+--+--+--|-------------------------------|---
  1 SSBN-726    | 1  1  1  1  1                                            |  5
  2 SSN-21      |         L1                                               |  1
  3 SSN-688     | 2  2  2  1  2                                            |  9
GDQ          #04|--+--+--+--+--+--+--+--+--|-------------------------------|---
  1 AE          |      Y1  1  1                                            |  3
  2 AG          |L1                                                        |  1
  3 AO-187     c|       1  1                                               |  2
                |--+--+--+--+--+--+--+--+--|-------------------------------|---
 14  33   TOTALS|29 24 33 30 27     1                                      |144
(?=help)
```

appropriation, award, start, keel, launch, or delivery. The page shown is using award as its basis milestone, so the assignment of two LSD-41's in fiscal 1986 indicates that two are projected to be awarded to Avondale that year.

It is fairly evident which yard and ship class each row is for, but how do you tell the job type and job series type? Notice the bottom row of assignments in the sample, those for AO-187's at General Dynamics' Quincy yard. There is a "c" in the last column of the class name area. This column holds a single character code which represents the job type. The "c" stands for "conversion", so the AO-187's are to be converted at Quincy. If there is a blank or an "n", that indicates new construction jobs. Other codes are "r" for reactivation and "s" for SLEP. A complete list of codes can be obtained from the assigner's on-line help.

The same yard, ship class, and job type will apply to all the ships assigned in a given row, but job series type must be individualized by ship. This is done with the single-character codes appearing in the table's cells. Notice the "L" in the fiscal 1988 column for LSD-49 new construction at Avondale---this indicates that one of the two ships awarded in that year will be a lead ship. Other codes are "F" for first follows and "Y" for yard leads. No code indicates an ordinary follow.

Notice the page number shown on the top line of the sample. The assigner display can be thought of as a window on a huge worksheet, with up to 500 rows and 260 columns (you can make assignments for 65 years in quarters, 20 years in months, five years in weeks, or 2/3 year in days). Up to twenty columns can fit on the display at a time. In the sample, the total period of interest is only nine years, so the columns required fill up less than a single page.

Moving the window around on the worksheet is known as "paging"; you can page right or left, and up or down. The page number label indicates both the horizontal and the vertical position of the current page. The numeric part indicates the vertical page, while the letter indicates the horizontal page.

The row and column totals which appear are the totals for the complete sheet, not just the page displayed; likewise the grand total at the bottom right gives the total number of assignments loaded. The totals at the bottom left (just to the left of the "TOTALS" label) are the number of yards and the number of assignments rows, respectively. Notice that each yard is labeled with a number; you can tell you are on the last vertical page if the number of the last yard is the same as the total number of yards.

8.3 ASSIGNER COMMANDS FOR WORKING WITH THE TABLE

Figure 8-4 summarizes the available assigner commands (it is part of the assigner's on-line help, and so can be referred to any time you are running the assigner). The commands fall into three categories: those for paging, those for changing the contents of the table, and miscellaneous service commands.

The assigner uses the same line-oriented prompt-and-response means of obtaining your commands as the Command System does. The "(?=help) >" appearing at the bottom of Figure 8-3 is its prompt. It will only accept one command per occurrence of the prompt.

8.3.1 Paging Commands

To see the next page to the right, give the ">" command. To see the farthest right-hand page, use ">>". The "<" and "<<" commands perform similar functions for movement in the leftward direction.

To see the next page down, use "+". The last page can be obtained by issuing "++". Upward paging is accomplished with "-" and "--".

Figure 8-4. Summary of Assigner Editing Commands

Command	Description	Command	Description
? =	Obtain help from a menu	? # =	Print help subject number #
=	Refresh assign display	Q =	EXIT assignments module
- =	Display previous page	-- =	Display topmost page
+ =	Display next page	++ =	Display last page
< =	Display left neighbor	<< =	Display leftmost page
> =	Display right neighbor	>> =	Display rightmost page
A =	Add a new yard	A # =	Add new shipclass to yard #
I # =	Add new yard before #	I #.@ =	Add new class before #.@
D # =	Delete an entire yard	D #.@ =	Delete class @ from yard #
MN # =	Modify Name of yard #	MN #.@ =	Modify Name of class #.@
P =	Display to line printer	P #,@ =	Print from yard # to @ on LP

M #.@ =	Modify assignments for class @ in yard #
R # =	Relocate yard numbered # to end of list
R #,## =	Relocate yard # to before yard number ##
R #.@,## =	Relocate #.@ to end of yard ##'s classes
R #.@,##.@@ =	Relocate class #.@ to before class ##.@@
RC # / RC #,## =	Like R, except copy yard instead of move
RC #.@,## / RC #.@,##.@@ =	Again like R; copy class instead of move

8.3.2 Editing Commands

The commands covered in this section are for addition, modification, and deletion of assignments. The basic commands are "A" for addition, "I" for insertion, "D" for deletion, and "M" for modification.

Most of these commands require you to specify which assignment row or shipyard you want them to act on. For example, if you wanted to modify the CG-47 assignments at Bath in Figure 8-3, you would give the command "M 2.1", meaning that you want to modify the first row in the second yard. You use the numbers with which each row and yard are labeled to indicate your selection. Note that these numbers can change as you add new assignments, so be sure to look at the display before giving a command.

When you ask to add assignments for a new yard you will be given prompts which resemble the following:

```
Give yard name please (Type "^" to cancel) > TODD LA

Make new assignment for yard #13, TODD LA

          Period:| 1  2  3  4  5  6  7  8  9

    Shipclass T|86 87 88 89 90 91 92 93 94

    ........... .. .. .. .. .. .. .. .. ..
    DDG-51         F1  2  2  2
```

You are expected to type the class name under the first section of dots. The appropriate job type code must be placed at the position of the last dot if the job is not of the new construction variety. Then type the number of assignments in each period (including series type codes, if any) in the pattern provided. You need not place each assignment at the position of the right-hand dot for each cell, but it is important not to put any numbers where there are blanks in the prompt. Keep in mind

that if you make a mistake you can always correct it using the Modify command.

--- Addition ---

To add an assignments row to a yard which has none, give the "A" command all by itself. You will be prompted for the name of the yard and then for the assignments.

To add to a yard which already has some, you may use either "A #", where "#" is the yard's number, or "I #.#", where #.# gives the yard and row within the yard you want the new row to precede (thus you can control the orde of display of class-jobs within a yard).

--- Modification ---

To modify a row of assignments you must specify the yard number and row within the yard to be changed (M #.#). A slightly different prompt than the one shown above will be given:

```
Modify assignment to yard BIW
(Blanks denote no change; use zero for assignment deletion)

        Period:| 1  2  3  4  5  6  7  8  9

    Shipclass T|86 87 88 89 90 91 92 93 94

  1 CG-47      | 1  1  1  1
               .. .. .. .. .. .. .. .. ..
                         2
```

The current values will be shown. You do not need to retype the whole line to make changes; instead, just space over under the cells to be changed and place the new values there.

You can change the names of yards or class-jobs using the "MN" (Modify-Name) command. If you specify only a yard number (M #) then you will be asked for the name of the yard those assign-

ments should be switched to; if you specify both a yard and row number (M #.#) then you will be prompted for a new class name and job type code.

--- Deletion ---

To delete all the assignments for a given yard, give the "D #" command, where "#" is the yard number. To delete only a given row within a yard, use "D #.#".

--- Moving Assignments Around on the Page ---

You can change the order in which assignments appear on the display page through use of the R (relocate) command. Its most general form is "R s.r,t.r", where "s.r" is the current yard and row number, and t.r is the yard and row number you would like the row to precede.

--- Copying Assignments ---

You can make a copy of a given row of assignments and place it under whatever yard and class-job name you please. The syntax of the RC command (relocate-copy) is just like that for the Relocate command (RC s.r,t.r). You will be asked for the new class name and job type code after giving the command.

8.3.3 Miscellaneous Service Commands

The assigner's service commands support the usual set of "housekeeping" functions. Help can be obtained with "?", which will bring up a menu describing the assigner's extensive on-line help. Make sure to look at this menu the first time you run the assigner so you will know what information is available.

The "&" command will cause the current page to be re-typed onto the display. Unless you have executed the assigner in "auto-refresh mode" (more on this in a moment), you will need to give the "&" command after giving the commands for changing to a different page.

The "Q" command terminates your editing session and starts the assigner's schedule creation and update pass. The "H" command allows you to temporarily exit back to the Command System, leaving your editing session on hold. You must come back and complete the session by giving the Q command if any changes you have made are to be saved. Note that changes you make to assigner parameters will have no effect on an editing session that is in progress.

The "P" command prints the contents of all assigner display pages, sending the output to the printer specified in the User Environment Parameter menu for your current scenario. The format of the output will be one display page per physical page.

8.4 ASSIGNER MODE CONTROL OPTIONS

You can vary a number of the assigner's characteristics, including the time span and time units used on the display page, the nature of the assignments loaded from the schedule data base, and the algorithms used to create new schedules during the update pass. You exercise this control by choosing values on the Command System's Assigner Parameter Menu, a sample of which is shown in Figure 8-5.

The parameters and their effects are:

1) TIME UNITS: This setting determines the span of time represented by each display page column. The smaller this span, the more precisely you specify the value of a given ship's basis date when you place its assignment in a given column.

2) STARTING DATE: The first date of interest to you for the next assigner run. The assigner will not load schedules with a basis date prior to this, nor will it permit you to make assignments prior to this date.

3) ENDING DATE: In combination with the starting date, determines the number of columns (periods) which will appear on the display page. You will not see nor be able to make assignments after this date.

Figure 8-5. Sample Assigner Parameter Menu

```
Menu is ASNPRM             * ALIAS COMMAND SYSTEM *   Scerario is DEMO
--------------------------------------------------------------------------
             MANUAL ASSIGNER MODULE INITIALIZATION PARAMETERS
 1.  TIME UNIT                = FISCYR        (FISCYR,CALYR,QTR,MONTH,WEEK,DAY)
 2.  STARTING DATE            =  1/ 1/1980    (MM/DD/YYYY)
 3.  ENDING DATE              = 12/31/1999    (MM/DD/YYYY)
 4.  CANDIDATE SHIP YARDS     = LIST          (ALL/LIST)
 5.  CANDIDATE SHIP CLASSES   = LIST          (ALL/LIST)
 6.  CANDIDATE JOB TYPES      = LIST          (ALL/LIST)
 7.  DISPLAY BASIS            = AWARD         (APPROP,AWD,START,KEEL,LNCH,DELIV)
 8.  ADJUST BASIS             = START         (APPROP,AWD,START,KEEL,LNCH,DELIV)
 9.  ADJUST MODE              = PROGRAM       (NONE,PROGRAM,COMPLX-GROUP)
10.  JOBS EPOCH OPTION        = PROJ          (ALL,CURR/PROJ,PROJ)
11.  SHIPCLASS SORT ORDER     = ALPHABETIC    (ALPHABETIC,INPUT ORDER)
12.  SHIPYARD SORT ORDER      = INPUT ORDER   (ALPHABETIC,INPUT ORDER)
13.  AUTO REFRESH             = OFF           (ON,OFF)

  COMMAND:
```

4) CANDIDATE SHIP YARDS These three parameters are

5) CANDIDATE SHIP CLASSES gates to list menus which

6) CANDIDATE JOB TYPES contain complete lists of all the yard names, ship class names, and job types which are known to your current scenario. The assigner will only load assignments for those yards, classes, and job types which are "turned on" in these list menus, and will only permit you to make assignments to those "turned on". Figure 8-6 contains a sample of the job type list menu.

   These lists are useful for restricting the volume of assignments you will have to work with in a given run. If you are mainly interested in assignments at one yard or for one class of ship, you can make settings in these list menus which will limit the assigner run to that yard or those classes. This reduces the amount of time you must spend paging around the assigner worksheet, and also reduces the overhead time spent loading and updating schedules.

7) DISPLAY BASIS: The milestone date you are implicitly specifying when you place an assignment in a given display page column, i.e. the basis date. You may choose any of the schedule milestones except commission.

8) ADJUST BASIS: The assigner's schedule creation logic can be instructed to construct schedules such that assignments are spread evenly over each period in a given yard. Suppose you assign four DDG-51's to be awarded to Ingalls in a given fiscal year. If the setting of this parameter is "START", the assigner can create four schedules in which the start dates are spread evenly over a twelve month period. If "KEEL", the keel dates will be so spread, and etc.

9) ADJUST MODE: This further controls the schedule creation logic. You can turn it off (i.e. have all four jobs' start dates be on the same day in the above example) by choosing "NONE"; date spreading can be done within each class by choosing "PROGRAM"; or it can be done for several classesa at once by choosing "COMPLX_GROUP". Each class is specified to belong to a given complexity group in the schedule job description relation ncjdat.descj. For example, if you choose COMPLX_GROUP and there are four DDG-51's and two CG-47's awarded to Ingalls in the same year, the six ships will be lumped together and their start dates spread evenly over a twelve month period.

Figure 8-6. Sample Assigner Valid Job Types List Menu

```
Menu is CHJTYP            * ALIAS COMMAND SYSTEM *   Scenario is DEMO
-------------------------------------------------------------------------
            CHOOSE THE SET OF VALID JOBS WHICH MAY BE ASSIGNED
-------------------------------------------------------------------------
   1. * CONV                              5. * REPAIR
   2. * NEWCON                            6. * SLEP
   3. * REACT                             7. * SLPCNV
   4. * REFUEL
-------------------------------------------------------------------------

 COMMAND:
```

10) JOBS EPOCH OPTION: This controls the loading of assignments from the schedule relations. To see historical, current, and projected assignments within your specified period you must choose "ALL"; to see only current and projected choose "CURR/PROJ"; and to see projected only choose "PROJ". Remember that any editing changes you make will affect only projected schedules. Substantial time can be saved during the load and update phases by choosing the PROJ option.

11) SHIPCLASS SORT ORDER: This controls the order in which class-job assignments rows are displayed within a yard. The choices are alphabetic and input-order, the latter meaning the order in which you added them to each yard (or the order you specified using the Insert and Relocate commands). Note that once you have chosen Alphabetic, the original input ordering will be lost.

12) SHIPYARD SORT ORDER: The assigner always orders yards alphabetically on the display at this time, regardless of this setting.

13) AUTO REFRESH: If "YES", the assigner will re-type the current display page after execution of each command. If "NO", you must give the "&" command whenever you want the display updated, including after issuance of paging commands. If you are operating on a low-baud rate terminal a choice of "NO" is likely to save you time, but you must keep more careful track of what you are doing since the effect of each command is not immediately displayed.

If you forget the parameter settings while in the middle of an assigner run and have a need to know them, they are accessible through the assigner's on-line help.

8.5 SCHEDULE CREATION AND UPDATE METHODOLOGY

After you issue the "Q" command, the assigner will create a set of schedules based on the contents of the display pages at that point, and will update the schedules in the ncjodat.projj relation to be consistent with them.

The creation is done by combining the information on the display page with that in the ncjdat.descj schedule planning factors relation. The basic task is to create as many schedules as there are new-construction-type jobs on the worksheet, and to fill in the fields in each such schedule. Referring to the

sample schedule record in Figure 8-2, the assigner can deduce the values for the SCENARIO, CLASS, YARD, NCJOBT, and JSTYP fields from what was showing on the display. It knows who you are and the current date, and so can fill in ENTRY_BY, ENTRY_DATE, and DATADATE. It starts by assuming that AUTOMOD is always "YES", always leaves SHIPNAME blank, and sets DATASOURCE to "assigner". It computes a value for ANSORDER based on the order of display of classes on the page. It deduces the basis date (say the basis milestone is AWARD for purposes of this discussion) from the column each assignment appeared in.

This leaves the other milestone date fields and some of the descriptive fields to be filled in. All but HULL are obtained by drawing on planning factor information in ncjdat.descj. A sample planning factor record is shown in Figure 8-7. The assigner's method is as follows. After it has filled in all the fields it can for a given schedule using only the display's contents, it looks in ncjdat.descj for the most appropriate job description/-planning factor record. It first looks for one (in the current scenario) which matches all of the values for CLASS, YARD, NCJOBT, and JSTYP. If it fails to find an exact match it tries for a match with a yard value of "ANY". If it still can't find a match it tries for one using a job series type of "ORDFCL" (ordinary follow). If this last attempt fails it will stop and require you to enter the appropriate planning factors.

Generally it will find an appropriate planning factor record. Using the values it finds there it can immediately fill in the COMNUM, CUSTOMER, CMETHD, and DAYSADDED fields. Then it uses the specified intervals between milestones to compute the rest of the milestone dates based on the value of the basis date.

The basis date will be set to the first day of the period whose column the assignment appeared in on the display, unless the time units used were fiscal or calendar years. In this case the year will be set according to the column, but the month and

Figure 8-7. Sample Job Description Records From Ncjdat.descj

```
$LINE SCENARIO      CLASS       NCJOBT YARD      JSTYP  COMNUM CMETHD CUSTOMER
COMPLEXGRP DEFLT DAYSADDED APPROPAWD AWDST STKL
KLLN LNDL DLCOM TIMUNT DATASOURCE    DATADATE ENTRYDATE ENTRYBY

   44 DEMO          LSD-41      NEWCON ANY       ORDFOL       1 MODULZ USN
11/01         0          1      12     6
   20     16      1 MONTHS 908          8/01/1984  8/02/1984 DBA
   45 DEMO          LSD-49      NEWCON ANY       ORDFOL       1 MODULZ USN
11/01         0          1      12     8
   23     24      1 MONTHS 908          8/01/1984  8/02/1984 DBA
   89 DEMO          LSD-49      NEWCON ANY       LEAD         1 MODULZ USN
11/01         0          1      12     8
   23     24      1 MONTHS 908          8/01/1984  8/02/1984 DBA
```

day will be according to the value of the DEFLTAWDAY field from the planning factor record.

If you have asked for date-spreading by setting the "ADJUST MODE" parameter on the parameter menu, this will be done. The algorithm basically gathers up all the ships to be spread, figures the total time over which the spreading is to take place, and divides this by the number of ships to obtain a standardized interval. It makes up a set of new notional milestones (for the adjustment date), recomputes what the display basis dates would have to be given the planning factor intervals, and then checks to make sure that the new basis dates all fall in the proper periods (display columns). If any don't, it shifts things around until they do, attempting to keep intervals as even as possible. The output of the algorithm is a new set of basis dates.

The assigner makes up new schedules for an entire yard at a time, and then updates ncjodat.projj with them. The actual update algorithm is complicated, but you can think of it as being equivalent to deletion of all the old schedules (if any) followed by addition of these new schedules, but with two provisions:

1) Schedules which were never loaded, i.e. those outside the period dealt with in the run, or those for classes, yards, or job types whose names were "off" in the assigner's list menus, will remain untouched. If you run the assigner, make no changes, and go through the update step, the number of schedules in ncjodat.projj will remain unchanged.

2) Schedules in the relation which have an AUTOMOD field value of "NO" will be untouched. Typically you will have set a value to "NO" in the DBU after making some changes that you want preserved (if you make changes and don't set AUTOMOD, the assigner is likely to replace/destroy your changes during your next assigner run). When the assigner finds such a "hard-wired" schedule, it just looks at its list of newly created schedules, throws out the one with the basis date closest to that of the "hard-wire", and goes on. This way the total number of schedules and their distribution over time remains correct.

The assigner _will_ delete "hard-wired" schedules if it has no new schedule for the same job in the same period, i.e. if you have deleted all the assignments in the period the "hard-wired" schedule is in.

Having updated the schedule relation's contents, the assigner must still set the values of the HULL field. It cannot do this prior to the update because HULL is part of the ncjodat.projj unique key field set, and an error would result if the assigner chose a hull number for a ship which was already assigned to another ship of the same class whose schedule was of the "hard-wired" variety.

The assigner attempts to assign realistic hull numbers; in order to do so, it will revise _all_ hull numbers in ncjodat. projj for the given scenario, not just those for the assignments loaded, except that it will not change hull numbers of "hard-wired" records. New hull numbers are assigned by class in order of delivery date, with the number for the ship with the earliest delivery being either:

1) One plus the largest hull number for ships of the same class appearing in the ncjodat.histj and ncjodat.currj relations, or

2) The number embedded in the ship's class name, if any (e.g., "51" for "DDG-51"), or

3) One.

The set of schedules which results, if loaded on a subsequent assigner run, will exactly duplicate the pattern of assignments they came from.

9.0 USING THE FORCE REPORT GENERATORS

ALIAS has two tools for projecting the impact on Navy force levels of projected new construction and repair programs, the Force Level Report Generator (FLRP) and the Battle Group Report Generator (BGRP). A sample of the output of the FLRP appears in Figure 9-1. The report gives the number of deployable ships available to the Navy in each of several future periods, by ship type. A sample Battle Group Report (Figure 9-2) provides the same fundamental information but in a more condensed format, computing the number of deployable battle group units available.

To execute the report generators, choose options 2 or 3 on the Command System's Force Level choice menu.

9.1 THE ALGORITHM

Both report generators base their results on the same force level computation algorithm. This searches each of the new-construction type schedule relations (ncjodat.histj, ndjodat. currj, and ncjodat.projj) for records of ships being commissioned. For each one found (in the classes of interest), it then searches the deact.miscj for a deactivation date. If one is found, the ship is added to the total number active during the given interval. If there is no specific deactivation date, a standard service lifetime, as specified in the shlife.miscj relation for the ship's class, is used to estimate the deactivation date. The estimate takes into account any additions to the standard life which the ship may enjoy due to work done in SLEP jobs, at reactivation, etc. (the algorithm searches for such jobs). It also takes the possibility of multiple commissionings and decommissionings (such as have occurred for the BB-61 class ships) into account.

After adding the ship to the number available in each appropriate period, a check is made for any repair-type jobs, for example SLEPs, which might have temporarily removed the ship from the deployable force level. If any are found, the total available in the appropriate periods is decremented.

Figure 9-1. Sample Force Level Report Generator Output

POM 86 FORCE IMPACT PROJECTION
BASED ON STANDARD SERVICE LIVES
(ALL DATA NOTIONAL)

CALENDAR YEAR		1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
CV	INVENTORY	8	8	8	8	8	8	7	7	8	7	7	8	8	8
CVN	INVENTORY	5	5	5	6	6	7	7	8	8	8	8	8	8	8
CVN	PROGRAM										1	1	1	1	1
CARRIER	DEPLOYABLE	13	13	13	14	14	15	14	15	16	16	16	17	17	17
CV	IN SLEP	1	1												
CARRIER	TOTAL	14	14	13	14	14	15	14	15	16	16	16	17	17	17
BB	PROGRAM			1	1	1	1	1	1	1	1	1	1	1	1
BB	TOTAL			1	1	1	1	1	1	1	1	1	1	1	1
CGN	INVENTORY	9	9	9	9	9	8	7	7	7	7	7	6	6	6
CG	INVENTORY	23	27	31	34	38	38	37	31	28	26	24	20	20	20
CG	PROGRAM						2	5	9	9	9	9	9	9	9
CRUISER	TOTAL	32	36	40	43	47	48	49	47	44	42	40	35	35	35
DDG	INVENTORY	37	37	37	38	38	38	38	38	38	38	38	38	38	38
DDG	PROGRAM						4	8	14	19	24	25	25	25	25
DDG	TOTAL	37	37	37	38	38	42	46	52	57	62	63	63	63	63
DD	INVENTORY	32	32	31	31	31	31	31	31	31	31	31	31	31	31
DD	TOTAL	32	32	31	31	31	31	31	31	31	31	31	31	31	31
FFG	INVENTORY	56	56	57	57	57	57	57	57	57	57	56	52	51	51
FFG	TOTAL	56	56	57	57	57	57	57	57	57	57	56	52	51	51
FF	INVENTORY	57	57	57	57	57	57	57	55	54	50	49	48	44	39
FF	TOTAL	57	57	57	57	57	57	57	55	54	50	49	48	44	39
FRIGATE	TOTAL	113	113	114	114	114	114	114	112	111	107	105	100	95	90
GRAND	TOTAL	228	232	236	241	245	251	255	258	260	259	256	247	242	237

Figure 9-2. Sample Battle Group Report Generator Output

DEPLOYABLE BATTLE GROUP PROJECTIFOR POM-86
BASED ON SURFACE COMBATANT REQUIREMENTS ONLY
(ALL DATA NOTIONAL)

CALENDAR YEAR	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
BATTLEGROUP														
CARRIER BG	10	10	10	11	11	11	12	13	16	16	16	17	17	17
SURFACE AG														
MARINE AF			1					1		1	1			
SUPPLY ESCORT	3	3				4	2	1	2			2	2	2
CONVOY	8	8	7	10	10	9	10	6	2					
BALANCE														
CARRIER	3	3	3	3	3	4	2	2						
BB														
1 1														
CRUISER														
DDG										4	4			
DD					1									
FFG														
FF	33	33	34	28	28	26	22	26	35	33	31	26	21	16

The result of the algorithm is a table of numbers available by period and ship class for the time span you specify as being of interest. The FLRP takes this table and formats it for output according to your instructions. The BGRP uses the table and a set of battle group "recipes" you specify to compute and format battle group availability.

Thus the data base data which will influence the contents of these reports are construction, conversion, and reactivation schedules (which determine force additions), schedules for some types of repair jobs (which determine temporary force subtractions), and deactivation dates and standard service lives (which determine permanent force subtractions (or perhaps temporary ones in the case of mothballings). The reports will thus immediately reflect changes which you make to projected new construction programs, and can also be used to study the impact of SLEP programs and changes in service lives due to improved maintenance cycles.

In addition to the raw data in the data base, you have two "handles" with which to control the contents and format of the reports. The first is the settings of the Command System's Force Level parameters; the second is the contents of report format control files.

9.2 PARAMETER OPTIONS

Figure 9-3 shows a sample Force Level parameters menu. The effect of the settings is as follows:

1) KEEP REPORT ON-LINE: The reports will appear on the printer you specify in the User Environment Parameters menu, but by setting this parameter to "YES" you can also have them saved into a disk file. You can then edit the disk file to make minor format modifications. They will be saved into a file called "FLREPT" in your log-on group.

2) REPORT START DATE: The start date of the period you wish the report to cover. This date should be the first day of a period; if it is not, it will automatically be set back to the first day.

Figure 9-3. Command System Parameter and List Menus Serving the Force Module

```
Menu is FLREPT              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
_________________________________________________________________________
              FORCE LEVEL AND BATTLEGROUP REPORT GENERATOR PARAMETERS
 1.  KEEP REPORT ON-LINE        = YES          (YES,NO)
 2.  REPORT START DATE          =  1/ 1/1986   (MM/DD/YYYY)
 3.  REPORT END DATE            = 12/31/2005   (MM/DD/YYYY)
 4.  RETIRE SHIPS BY            = LIFE         (LIFE,DATE)
 5.  TIME PERIOD LENGTH         = CALYR        (DAY,WEEK,MONTH,QTR,CALYR)
 6.  IN FORCE DAY               = END          (BEGIN,END)
 7.  PROGRAM MILESTONE          = APPROP       (APPROP,AWD,DELIV)
 8.  OUT OF FORCE REPAIR JOBS   = LIST         (ALL/LIST)

  COMMAND:
```

```
Menu is FLREPT              * ALIAS COMMAND SYSTEM *    Scenario is DEMO
_________________________________________________________________________
         REPAIR JOBS THAT REMOVE A SHIP FROM FORCE DURING EXECUTION
_________________________________________________________________________
    1.   REFUEL                         3. * SLEP
    2.   REPAIR                         4.   TESTRE
_________________________________________________________________________

  COMMAND:
```

3) REPORT END DATE: The last day of the period you wish the report to cover. Note that due to printer width limitations, the maximum duration a report can cover is 20 periods. If you ask for more, the end data will be set back to give a 20 period duration.

4) RETIRE SHIPS BY: This allows you to control how the algorithm determines retirement dates. If DATE, it use a deactivation date for a given ship if it can find one. If LIFE, it will use the deactivation date only if the date is prior to the date of the run (i.e. the ship has already retired). This parameter can be useful for studies in which you want to assess the effect of a wholesale change in service lives; it makes it unnecessary for you to change the deactivation dates for currently active ships one at a time.

5) TIME PERIOD LENGTH: Duration a report column represents. Used in combination with the start and end dates to determine the number of columns appearing on a report.

6) IN FORCE DAY: Most ships will retire in the middle of a given period; there must be a convention for deciding whether a ship is or is not to be counted in the force during that period. There are two choices for the convention: if BEGIN a ship is counted in the force if it was active on the first day of a period; if END it is counted only if active on the last day of a period.

7) PROGRAM MILESTONE: Notice in Figure 9-1 that the numbers of deployable ships are broken out into the categories of "inventory" and "program". You have the capability to break down the numbers of each ship type available into up to five separate categories on the basis of their date of entry into the force. This allows you to assess the effect of different parts of a construction program (e.g., a POM and an EPA) with more precision. This parameter specifies which ship construction milestone is to provide the basis date for determining which category a ship falls into.

8) OUT OF FORCE REPAIR JOBS: This parameter is a gate to a list menu, also shown in Figure 9-3, which contains the names of repair job type codes. Those jobs which are "turned on" in the list menu will be considered by the force computation algorithm to remove a ship temporarily from the force while they are being done.

9.3 REPORT CONTROL FILES

In the sample reports, ship types and battle groups appear in a particular order. You have control over this order and over

which ship classes are part of each type, and which can help make up each kind of battle group. You exercise this control by providing the report generators with a format control file specifying what you want done. These files required for report generator operations.

The files contain commands and data values. There are two types of file (i.e. two different sets of commands), one for each kind of report.

9.3.1 Force Level Report Control File Syntax

A sample report control file suitable for FLRP appears in Figure 9-4. The file contains three kinds of lines, none of which may exceed 72 characters in length:

1) COMMENT LINES: These are lines which begin with a "%". They are ignored by FLRP. You may type comments and notes to yourself on these lines to remind yourself of your rationale for the file's contents. FLRP will also ignore blank lines.

2) COMMAND LINES: These begin with one of the nine valid keywords, and typically have additional information following the keyword such as ship class names and labels which are to appear on the printout. Such information must always be separated from the keyword by one or more spaces. If there are multiple specifications, e.g. multiple class names, they must be separated from each other by commas.

3) CONTINUATION LINES: These begin with a "+", and indicate to FLRP that you could not fit all the information you needed to onto the previous keyword line.

The valid keywords and their meanings are:

PRGLB Stands for "program label specification". Following the keyword a label to appear on output and a date in MM/DD/YYYY format are required. All ships whose "program milestone" (see the discussion of parameters in the previous section) is after the given date but not before the date given on the <u>next</u> PRGLB line (if any) will appear on the report on lines having the given label. PRGLB lines must be the first commands in the file, and must appear in the order of their dates. They are how you break out force

Figure 9-4. Sample Force Level Report Format Control File

```
% THIS IS A FORCE LEVEL REPORT FORMAT CONTROL FILE
% ANY LINE BEGINNING WITH % IS CONSIDERED A NOTE AND IGNORED
%
%     The next two lines tell FLRP to split the force level
%     into two lines for each class, based on ship approp date
PRGIB   Inventory,1/1/1900
PRGIB   Program  ,10/1/1986
%
%     The TITLE lines give the title that will be printed
%     (centered) on the top of each report page
TITLE POM 86 Force Impact Projection
TITLE Based on Standard Service Lives
TITLE (All Data Notional)
%
%     Start the report specification.  BTOT lines tell FLRP to
%     start keeping a running total, ETOT where in the body
%     to print the total; last two words on ETOT lines are
%     the left and right labels actually printed.  Label
%     on BTOT line and first one on ETOT for internal FLRP use.
%     EITOT is like ETOT except specifically designed for
%     subtotals; it ensures no page feed happens in the
%     middle of a type being printed.
%     TYPE lines specify ship types by
%     giving the names of all the classes in the type.

START
BTOT grand
BTOT subn
TYPE SSBN,     SSBN-726,SSBN-640,SSBN-627,SSBN-616,
+              SSBN-611,SSBN-610,SSBN-609,SSBN-601,SSBN-599,
+              SSBN-598
ETOT subn,SSBN,TOTAL
BTOT sub
TYPE SSN,      SSN-688,SSNX,SSN-21,SSN-575,SSN-578,SSN-585,SSN-594,
+              SSN-597,SSN-637,SSN-671,SSN-685
ETOT sub ,SSN,TOTAL
%
%     note job line causes carriers in SLEP to be printed
%     they do not appear in the deployable total
BTOT carrier
BTOT dcarrier
TYPE CV,       CV-41,CV-59,CV-63,CV-67
TYPE CVN,      CVN-65,CVN-68
EITOT dcarrier,CARRIER,DEPLOYABLE
JOB CV,        IN SLEP,SLEP,      CV-41,CV-59,CV-63,CV-67
ETOT carrier,  CARRIER,TOTAL
%
BTOT bb
TYPE BB,       BB-61
ETOT bb,       BB,TOTAL
```

Figure 9-4. Sample Force Level Report Format Control File

```
%
BTOT cruisers
TYPE CGN,       CGN-25,CGN-36,CGN-38,CGN-35,CGN-9
TYPE CG,        CG-16,CG-26,CG-47
ETOT cruisers,  CRUISER,TOTAL
%
BTOT ddg
TYPE DDG,       DDG-2,DDG-37,DDG-51,DDG-993
ETOT ddg,       DDG,TOTAL
%
BTOT dd
TYPE DD,        DD-945,DD-963
ETOT dd,        DD,TOTAL
%
BTOT frigate
BTOT ffg
TYPE FFG,       FFG-1,FFG-7
ETOT ffg,       FFG,TOTAL
BTOT ff
TYPE FF,        FF-1037,FF-1040,FF-1052
ETOT ff,        FF,TOTAL
ETOT frigate,   FRIGATE,TOTAL
%
BTOT amphibs
TYPE AMPHIBS,   LHD-X,LSD-41,LSD-49,LCC-19,LHA-1,LHD,LHD-1,
+               LKA-113,LPD-1,LPD-4,LPH-2,LSD-28,LSD-36,LSD-41,
+               LSD-49,LST-1179
ETOT amphibs,   AMPHIBS,TOTAL
%
BTOT mine
TYPE MINE CM,   MCM-1,MSH-1,MSO-422,MTS
ETOT mine,      MINE SHIPS,TOTAL
%
BTOT aux
TYPE AUXILIARY,AD-37,AD-41,AE,AE-21,AE-23,AE-26,AF-58,AFDM,
+               AFS-1,AG,AK-286,AO-143,AO-177,AO-187,AO-51,
+               AOE,AOE-1,AOR-1,AR,ARDM-4,ARS-50,AS-19,AS-31,AS-33,
+               AS-36,AS-39,ASR-21,ATF-166,ATS-1
ETOT aux,       AUXILARY,TOTAL
%
BTOT T-SHIPS
TYPE T-SHIPS,   T-ACS,T-AG,T-AGOS-1,T-AO-187,T-ARC-7,T-AVB,TAGOS-1,
+               TAH-X,TAO-187
ETOT T-SHIPS,   T-SHIPS,TOTAL
%
ETOT grand,     GRAND,TOTAL
STOP
```

availability by program source; note in the sample in Figure 11-4 that all ships appropriated before fiscal 1987 will be classes as inventory ships, while all appropriated later will be classed as POM ships.

TITLE These lines must appear after the PRGLB lines but before any other keyword. They are used to specify the report's title, which will be centered over the body of the report. Up to five are allowed, and you may specify blank TITLE lines to leave blank lines in the title text.

START FLRP will ignore any keywords which occur before a START line, or which appear after a STOP and before the next START. This allows you to make up a large standardized report control file and then use parts of it selectively. For example, if you did not want information on carrier availability to print out, you could place a STOP line just before the section of the file dealing with carriers, and a START line right after it. It is not necessary to delete lines from the control file to effectively remove them.

STOP Used in conjunction with START. Any lines read after a STOP is encountered but before the next START will be ignored.

TYPE This is in some sense the most important keyword. TYPE lines give the label for a section of FLRP output, followed by a list of ship classes whose availability is to be added up to give the availability figures for that section (where a section is defined as, e.g., the program and inventory lines). In the sample file, it is specified that on the output classes CVN-65 and CVN-68 are to be lumped together under the label "CVN". You may specify as many or as few classes per type as you desire. However, it is very important that you specify only valid class names for your scenario. The computation algorithm logic is such that it will simply ignore any invalid class names, perhaps then giving you results which do not include all the classes you think they do. The order of the TYPE lines in the file controls the order in which their corresponding sections will appear on the report.

JOB This keyword lets you have the number of ships undergoing a given repair job appear on the report. The syntax of the line is "keyword label job_type_code class_list". The label will appear on the report; the job type code tells FLRP which job you mean; and the class list which classes you wish totalled on the report line. The most common use of the keyword is to provide a tally of the number of carriers in SLEP, in combination with a parameter specification which says that ships undergoing SLEP are to be considered undeployable. The deployable carriers figure and the in-SLEP figure can then be combined to arrive at a figure for total carriers in each period.

BTOT Stands for "begin totaling", and must be followed by a total name. The command causes FLRP to establish a totaling buffer line to which the contents of all TYPE and JOB sections are added until processing encounters an ETOT or EITOT line, at which point the total is printed. By placing a BTOT before a TYPE line and an ETOT following it, you can have the inventory and program lines added up on the output to provide a total-deployable measure for the TYPE. The commands can likewise be used to for computation of more extensive subtotals and for grand totals.

ETOT Stands for "end totaling". Must have a total name which matches a name on a previous BTOT line. If any other BTOT lines were given between that BTOT and this ETOT, they must have matching ETOT's before this one as well (i.e. BTOT's and ETOT's can be nested, but the nesting must be strict). After the name two labels must appear, one for the report's far left label column and one for its right-hand label column.

EITOT Stands for "end internal total". This has the same effect as ETOT, but forbids page breaks from occurring (FLRP will start a new page only following an ETOT line; if necessary it will move an entire section onto the next page).

When constructing an FLRP format control file, keep in mind that the order in which the keywords appear in the file matters. In general, the order should be PRGLB's, TITLE's, and START's, followed by sets of BTOT's, TYPE's, JOB's, and ETOT's or EITOT's. Publicly available format control files are stored in the .fmtfil group. The sample file discussed here is stored in that group; you may use it as a basis for any you contruct.

9.3.2 Battle Group Report Format Control File Syntax

A sample format control file suitable for BGRP appears in figure 9-5. It is similar in basis outline to those for FLRP, being composed of comment lines, keyword command lines, and continuation lines. The keywords are not the same, however, since they support a different logic.

BGRP expects you to define a set of kinds of battle group which the Navy desires, along with a target number for each kind. It also expects you to tell it what sorts of ship can be used to make up each group, along with recipes of the mixtures required.

Figure 9-5. Sample Battle Group Report Format Control File

```
% ALIAS BATTLE GROUP REPORT FORMAT/CONTENTS DEFINITION FILE
% format is: title; start; type; function; bgroup; makeup; end
% title line has titles for report
% start line indicates start of processing
% type line indicates ship classes making up a type
% function line lists types which can perform a function, in
%  order of preference
% bgroup  describes battle groups to be made up
% makeup  describes which functions each battle group requires
%
TITLE Deployable Battle Group Projection For on POM-86
TITLE Based on Surface Combatant Requirements Only
TITLE (All Data Notional)
%
START
% type format similar to force level report: name,label,class list
%
TYPE CARRIER,   CARRIER,    CV-41,CV-59,CV-63,CV-67,CVN-65,CVN-68
TYPE BB,        BATTLESHIP,BB-61
TYPE CRUISER,   CRUISER,    CGN-25,CGN-36,CGN-38,CGN-35,CGN-9,CG-16,CG-26,CG-47
TYPE DDG,       DDG,        DDG-2,DDG-37,DDG-51,DDG-993
TYPE DD,        DD,         DD-945,DD-963
TYPE FFG,       FFG,        FFG-1,FFG-7
TYPE FF,        FF,         FF-1037,FF-1040,FF-1052
%
% function format is name,list of types which can perform it
%  in order of preference
FUNCTION CRUISER, CRUISER,BB
FUNCTION CARRIER, CARRIER
FUNCTION DDG,     DDG,CRUISER,BB
FUNCTION DD,      DD
FUNCTION FRIGATE, FFG,FF
%
% bgroup format is name,output label,priority,target level,
%  begin date this defn takes effect, end date this defn effective
BGROUP CVBG,CARRIER BG,1,17,1/1/1900,1/1/2111
BGROUP SAG,SURFACE AG, 3, 4,1/1/1900,1/1/2111
BGROUP MAF,MARINE AF,  2, 2,1/1/1900,1/1/2111
BGROUP ESC,SUPPLY ESCORT,4,10,1/1/1900,1/1/2111
BGROUP CON,CONVOY,     5,10,1/1/1900,1/1/2111
%
% makeup format is battle group name, function, # reqd, func, #reqd
MAKEUP CVBG,    CARRIER,1,CRUISER,1,DDG,4,DD,2,FRIGATE,4
MAEKUP SAG,     CRUISER,2,DDG,2,FRIGATE,2
MAKEUP MAF,     CRUISER,2,DDG,2,DD,4,FRIGATE,10
BGROUP ESC,     DDG,1,DD,1,FRIGATE,2
BGROUP CON,     DD,1,FRIGATE,4
STOP
```

The recipes are called MAKEUPs; they are composed of ship FUNCTIONs, which can be performed by one or more ship TYPEs, and the TYPEs are in turn made up of one or more classes.

Given this information and the table of per-period availabilities by ship class produced by the basic force computation algorithm (see Section 9.1), BGRP attempts to fill the battle groups in each period out of available ships and in priority order. Its method is to treat one period at a time, and to "build" battle groups of the highest priority until it either hits the target for that battle group or runs into a shortage. It then goes on to the battle group next in priority. When searching for ships which can perform a given FUNCTION, it will use draw from TYPEs in the order in which they are named on the FUNCTION line in the file, so TYPE's should appear on those lines in the order of their appropriateness for fulfilling the function.

The keywords are:

TITLE — Similar in function to the TITLE keyword in FLRP files; you can use it to specify up to five lines of report title text, which will be centered.

START
STOP — The role of these keywords is identical to the one that they play in FLRP files. See the preceding Section.

TYPE — These group ship classes into ship types. Following the keyword must be the name of the type, a label, and then the list of classes belonging to that type. As with FLRP files, if the class list is too long for one line, it may be continued on the next line by placing a "+" in the first column. The label is used in the "BALANCE" section of the report, which gives the number of ships of each type remaining after BGRP has done its best to make up the battle groups.

FUNCTION — These lines define functional categories of ships, and are the terms in which battle group recipes are specified. Following the keyword must be a name for the given function and then a list of type names, as defined on previous TYPE lines.

BGROUP Battle GROUP lines define the battle groups to be made up, one per line. The format of the specification is "keyword group_name,group_label,target_number, target_effective_date, target_end_date". The label will appear on the report, marking the line for the given group. Note that the groups appear on the report in the order in which they are specified in the file. They target number is effective during the period specified by the dates. It is permitted to have more than one line for a given battle group (they must have the same name so they can be associated) in order to change the target sometime during the report period.

MAKEUP These lines give the recipes for each battle group, one line per group. The name of the battle group being defined must appear immediately after the keyword (it must have been defined by a previous BGROUP line). Following this, pairs of "function, number" must be specified, each of which indicates that the given battle group requires that number of that function.

The keywords must appear in a strict order in BGRP format control files: all TITLE lines, followed by all TYPE lines, followed by all FUNCTION lines, followed by all BGROUP lines, followed by all MAKEUP lines. START and STOP lines can be used to "disconnect" other lines, but you must be careful not to disconnect any lines which are implicitly required by lines farther down the file (unless you disconnect those lines as well).

END

FILMED

3-85

DTIC